

Informatische Grundkonzepte in Klasse 5 der Realschule

—

Entwurf und Evaluation von fächerübergreifendem Unterricht

Dissertation
zur Erlangung des Grades einer Doktorin
der Erziehungswissenschaften (Dr. paed.)
der Pädagogischen Hochschule Ludwigsburg

vorgelegt von Birgit Wursthorn aus Reutlingen

Ludwigsburg
2006

Erstgutachter: Prof. Herbert Löthe
Pädagogische Hochschule Ludwigsburg

Zweitgutachter: Prof. Dr. Andreas Zendler
Pädagogische Hochschule Ludwigsburg

Datum des Abschlusses der mündlichen Prüfung: 18.05.2006

Iucundi acti labores

(Cicero)

Danksagung

Eine Forschungsarbeit in der Schule wird von sehr vielen Menschen beeinflusst, insbesondere wenn sie sich über ein ganzes Schuljahr erstreckt und so massiv in den Unterricht eingreift wie in dieser Arbeit. Die Voraussetzungen für einen reibungslosen Ablauf schaffte der Schulleiter Herr Petri. Der wichtigste Erfolgsfaktor waren jedoch die Schülerinnen und Schüler. Ihren Einsatz, ihre Mitarbeit, ihr Engagement und ihr Durchhaltevermögen möchte ich besonders hervorheben. Benennen möchte ich auch diejenigen Eltern, die die Arbeit durch eine positive Haltung gegenüber ihren Kindern unterstützten. Wichtig war mir auch das Gefühl, bei den Lehrerinnen und Lehrern des Kollegiums willkommen zu sein. Allen genannten Personen möchte ich ganz herzlich danken.

An der Pädagogischen Hochschule Ludwigsburg gilt mein besonderer Dank meinen Betreuern Prof. Herbert Löthe und Prof. Dr. Andreas Zendler, die mir in den verschiedenen Phasen der Arbeit hilfreich zur Seite standen und Anregungen gaben.

Auch meine Kolleginnen und Kollegen am Institut für Mathematik und Informatik der Pädagogischen Hochschule Ludwigsburg spielten eine wichtige Rolle. Besonders hervorheben möchte ich Dr. Rose Vogel, Christian Spannagel und die ehemalige Kollegin Prof. Dr. Christine Bescherer für ihre geschätzten Anstöße in der Endphase. Vielen Dank auch Frau Ertelt und Frau Hübner für das intensive und gründliche Lesen der Arbeit.

Außerdem danke ich meinen Eltern, die mich schon sehr früh förderten und somit den Grundstein für meinen Werdegang legten. Nicht unerwähnt soll auch meine kleine Nichte Sonja bleiben, da sie es in der schwierigen Zeit des Abfassens der Arbeit am besten schaffte, meine Gedanken von der Arbeit abzulenken, und eine ständige Quelle der Freude war.

Mein größter Dank gilt den am Unterrichtsversuch beteiligten Lehrerinnen Frau Hartlieb, Frau Bräutigam und Frau Rupp. Bereits vor dem Unterrichtsversuch waren sie bereit, sich in zahlreichen Sitzungen in das Unterrichtskonzept einzuarbeiten. Während des Schuljahrs ließen sie mir den für die Forschungsarbeit notwendigen Freiraum und reagierte auf teilweise sehr kurzfristige Wünsche sehr flexibel. Für meine Probleme hatten sie stets ein offenes Ohr und aufgrund ihrer langjährigen Praxis überaus hilfreiche und für mich sehr produktive Ratschläge. Insbesondere ihre Verlässlichkeit, aufgeschlossene Freundlichkeit und ihre Ermutigung stärkten mich in meiner Arbeit und ermöglichten mir diese wunderbare und unvergessliche Zeit in der Schule.

Inhaltsverzeichnis

1	Einleitung	1
2	Informatische Bildung	5
2.1	Empfehlungen zu einer informatischen Bildung	5
2.2	Bildungspläne der Sekundarstufe I in Baden-Württemberg	9
2.2.1	Historische Entwicklung	9
2.2.2	Bildungsstandards für Informationstechnische Grundbildung an Realschulen	12
2.3	Standpunkte der Fachdidaktik Informatik	13
2.4	Informatische Grundkonzepte in Klasse 5 der Realschule	21
3	Konzept des Unterrichtsversuchs	29
3.1	Logo	31
3.1.1	Logo-Philosophie	32
3.1.2	Logo als Programmiersprache	38
3.1.3	Logo und informatische Grundkonzepte	44
3.2	Rahmenbedingungen	47
3.3	Fachinhalte und informatische Grundkonzepte	53
3.4	Unterrichtsmethoden	55
3.5	Vorbereitung der Fachlehrerinnen	58
3.6	Vorbereitung der Eltern	60
4	Unterrichtseinheiten zu informatischen Grundkonzepten	61
4.1	Fachunterricht	61
4.2	Projektarbeit	92
4.3	Fachunabhängiger Unterricht	96
5	Konzeption und Durchführung der Untersuchung	103
5.1	Fragestellung und Untersuchungsdesign	103
5.2	Quantitative Erhebung	104
5.2.1	Testentwicklung zu informatischen Grundkonzepten	105
5.2.2	Test zu informatischen Grundkonzepten	108
5.2.3	Kognitiver Fähigkeitstest	108
5.2.4	Fragebogen zu Einstellungen zum Computer	109
5.3	Qualitative Erhebung	110

5.3.1	Forschungstagebuch	110
5.3.2	Leitfadeninterview Lehrerinnen	112
5.3.3	Leitfadeninterview Schülerinnen und Schüler	113
5.3.4	Unterrichtsmaterialien zu einzelnen Unterrichtseinheiten	114
6	Darstellung und Interpretation der Ergebnisse	115
6.1	Gesamtkonzept	116
6.1.1	Verfahren zur Auswertung der Interviewdaten	116
6.1.2	Aspekte des Gesamtkonzepts	120
6.1.3	Lernerfolge allgemein	126
6.1.4	Verhalten der Schülerinnen und Schüler im Unterricht	130
6.1.5	Rahmenbedingungen	135
6.2	Projektarbeit	138
6.3	Lernerfolg informatischer Grundkonzepte – Vor- und Nachtest	142
6.3.1	Informatische Grundkonzepte allgemein	144
6.3.2	Informatische Grundkonzepte und Geschlecht	145
6.3.3	Informatische Grundkonzepte und Fähigkeiten	146
6.4	Lernerfolg informatischer Grundkonzepte – Unterrichtsmaterialien	150
6.4.1	Wirkprinzipien von Informatiksystemen	150
6.4.2	Informatische Modellierung	159
6.4.3	Arbeitstechniken	161
6.4.4	Informatische Beschreibungsmittel	167
6.4.5	Fachnahe Inhalte	174
6.4.6	Zusammenfassung und Interpretation	176
6.5	Veränderung der Einstellungen zum Computer	178
7	Fazit	189
7.1	Zusammenfassung	189
7.2	Grenzen der Untersuchung	192
7.3	Konsequenzen für die Implementierung des Unterrichtskonzepts	193
Anhang		197
A	Empfehlungen zu einer informatischen Bildung	197
B	Informatische Grundkonzepte – Vortest	200
C	Einstellungen zum Computer – Fragebogen	224
D	Rücklauf der Fragebögen zu den Einstellungen zum Computer	228
E	Interviewleitfaden Lehrerinnen	231
F	Interviewleitfaden Schülerinnen und Schüler	236
G	Informatische Grundkonzepte – Tests im laufenden Schuljahr	239
H	Schülerbeispiele	249
	Abbildungsverzeichnis	250

Tabellenverzeichnis	254
Literaturverzeichnis	256

1 Einleitung

Es ist unbestritten, dass Fähigkeiten im Umgang mit Informatiksystemen bereits heute in vielen Bereichen zur praktischen Lebensbewältigung hilfreich oder sogar notwendig sind. Das diesen Fähigkeiten zu Grunde liegende Wissen ist allgemein bildend und muss in der Schule vermittelt werden. Seit 1970 erfolgt dies bereits auf Basis verschiedener Ansätze. In den siebziger Jahren stand inhaltlich die Hardware im Vordergrund. Dann rückten mehr und mehr Algorithmen und Anwendungen in den Mittelpunkt. In den achtziger Jahren konzentrierte man sich auf die Benutzung von Anwendungssystemen, um lebenspraktische Orientierung zu vermitteln. In den neunziger Jahren wurde die Benutzerorientierung von der Wissenschaft vom Entwurf und der Gestaltung von Informatiksystemen abgelöst (Baumann 1996, S. 112 ff., Hubwieser 2001, S. 50 ff.). Eine grundlegende informatische Bildung als Voraussetzung für die kompetente, effiziente und verantwortungsvolle Nutzung des Computers umfasst sicherlich Aspekte aus allen Ansätzen. In ihr müssen nach Meinung von Engbring (2003, S. iii) die Inhalte der Kerninformatik, d.h. der theoretischen, technischen und praktischen Informatik, ausgehend von konkreten Anwendungssystemen, eingeführt werden. In diesem Punkt stimmen die verschiedenen Ansätze der Fachdidaktik Informatik, wie sie in Kapitel 2.3 beschrieben werden, überein. Strittig ist häufig jedoch der Zeitpunkt, zu dem die Einführung in die informatische Bildung an Schulen beginnen soll.

“Dies sollte frühestens in der 6. Jahrgangsstufe geschehen, da die Schüler erst in diesem Alter die nötige Abstraktionsfähigkeit erlangen, um über das konkret verwendete System hinaus allgemeinere, grundlegende Konzepte zu verstehen.“ (Hubwieser, 2001, S. 98)

Damit verweist Hubwieser auf Piaget (Fatke, 2003, S. 66), nach dessen Meinung aussagenlogische oder formale Operationen, die Voraussetzung für viele informatische Inhalte sind, erst im Alter zwischen 11 und 13 Jahren erbracht werden können. Schwill (2001) hingegen behauptet, dass einige wichtige fundamentale Ideen der Informatik bereits von Grundschulkindern erlernt werden können, „vorausgesetzt, die Gegenstände werden altersgemäß aufbereitet und im Unterricht unter Berücksichtigung der bis zu diesem Zeitpunkt aufgebauten kognitiven Strukturen der Kinder und unterstützt durch Handlungen oder reale Gegenstände vermittelt“ (vgl. Schwill 2001, S. 29). Dies zeigt er für die fundamentalen Ideen der Rekursion, der Greedy-Methode und der strukturierten Zerlegung. Neben entwicklungspsychologischen Argumenten werden gegen einen zu frühen Einstieg in eine informatische Bildung auch organisatorische Gründe angegeben.

Hubwieser (2001, S. 97) zum Beispiel, warnt vor einer Stofffülle in Klassenstufe 5, fordert jedoch einen möglichst frühen Beginn.

Die aktuellen fachdidaktischen Forschungsarbeiten beschäftigen sich hauptsächlich mit Konzepten für die Sekundarstufe II. Humbert (2003) entwirft ein Modulkonzept für die Themenbereiche *Informatiksysteme verantwortlich nutzen*, *Elemente der theoretischen Informatik* und *informatische Modellierung* für den Anfangsunterricht Informatik in der Sekundarstufe II. Das von Schulte (2003) entwickelte *life*³-Unterrichtskonzept baut auf einem objektorientierten Zugang zur Informatik im Anfangsunterricht in der Jahrgangsstufe 11 auf. Die Arbeit von Modrow (2003), in der die fundamentalen Ideen als Leitlinien zur Curriculumentwicklung herangezogen werden, bezieht sich fast ausschließlich auf das Schulfach Informatik der Sekundarstufe II. Brinda (2004) stellt in seiner Dissertation ein didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II vor. Die von Reichert (2003, S. 61) entwickelten Kara-Programmierungsumgebungen werden an Universitäten in der Lehrerbildung, an Fachhochschulen und Gymnasien eingesetzt. Sie scheinen also nicht für den Unterricht zu Beginn der Sekundarstufe I geeignet zu sein. Auch die Beiträge auf der GI¹-Fachtagung „Informatik und Schule 2003“ beschäftigen sich hauptsächlich mit Unterrichtsansätzen am Ende der Sekundarstufe I oder der Sekundarstufe II. Eine Ausnahme bildet allein der Vortrag von Frey (2003, S. 33), der den in der Jahrgangsstufe 6 neu eingeführten Informatikunterricht an bayrischen Gymnasien vorstellt.

Nach Ansicht der Autorin beschäftigen sich diese Ansätze zwar mit relevanten und wichtigen Konzepten der Informatik, der Zeitpunkt ihrer Einführung ist jedoch zu spät. Das informatische Grundlagenwissen muss und kann nach ihrer Meinung früher vermittelt werden, so dass es spiralförmig in weiteren Klassenstufen auf unterschiedlichen Abstraktionsniveaus wiederholt und vertieft werden kann. Aus diesem Grund wird in der vorliegenden Arbeit ein Unterrichtskonzept zur Vermittlung von informatischen Grundkonzepten in Klasse 5 der Realschule entwickelt. Um die informatischen Grundkonzepte in Anwendungskontexte einzubetten, wird der Unterricht in die Fächer Mathematik, Englisch, Deutsch und Musik integriert und findet in der konkreten Umsetzung vier Stunden in der Woche über ein gesamtes Schuljahr hinweg statt. Dadurch muss in der Stundentafel kein eigenes Fach ausgewiesen werden. Der Unterricht wird bereits, wie im Unterricht zur *Informationstechnischen Grundbildung* im Bildungsplan 2004 in Baden-Württemberg gefordert, in die verschiedenen Fächer integriert. Die Programmiersprache Logo wird in diesem Konzept als pädagogisches Werkzeug eingesetzt, um die informatischen Grundkonzepte für die Schülerinnen und Schüler altersgemäß aufzubereiten und ihnen die Möglichkeit anzubieten, mit den teilweise theoretischen Konzepten konstruktiv zu arbeiten und eigene Produkte am Computer zu erstellen. In den vier Unterrichtsstunden werden also informatische Grundkonzepte, Fachinhalte und Logo in teilweise fächerübergreifenden Szenarien erlernt. Wie die konkrete Erprobung zeigte (siehe Kapitel 4), stellte

¹Gesellschaft für Informatik

dies eine große Herausforderung sowohl an die Schülerinnen und Schüler als auch an die Forscherin als Lehrperson und Entwicklerin des Konzepts dar. Sie wurde aber von allen Beteiligten bewältigt, was die Evaluation in Kapitel 6.1.2 zeigt.

In Kapitel 2 wird zunächst die Entwicklung der in Deutschland verabschiedeten Empfehlungen zu einer informatischen Bildung skizziert. Es schließt sich die Analyse der Bildungspläne der Sekundarstufe I in Baden-Württemberg im Hinblick auf ihre informatischen Inhalte an. Außerdem werden die verschiedenen Positionen der Fachdidaktik Informatik vorgestellt. Die in dieser Arbeit für einen Einstieg in die informatische Grundbildung in Klassenstufe 5 der Realschule ausgewählten informatischen Grundkonzepte werden in Kapitel 2.4 vorgestellt und u.a. auf Basis der Empfehlungen, Bildungspläne und fachdidaktischen Positionen begründet. Die Entwicklungsarbeit für das Unterrichtskonzept zur Vermittlung der informatischen Grundkonzepte wird in Kapitel 3 dargestellt. Besondere Aufmerksamkeit erhalten die Aspekte der Logo-Philosophie und die Spracheigenschaften von Logo, die meines Erachtens im besonderen Maße das Erlernen der informatischen Grundkonzepte unterstützen. Außerdem werden die verschiedenen Rahmenbedingungen, die Einbettung der informatischen Grundkonzepte in die Fachinhalte der Fächer Mathematik, Englisch, Deutsch und Musik der Klassenstufe 5 und die eingesetzten Unterrichtsmethoden präsentiert. Es schließt sich die Vorbereitung der Fachlehrerinnen und der Eltern an. Die einzelnen Unterrichtseinheiten stehen in Kapitel 4 im Zentrum. Es werden hier die drei Phasen, in die sich die Vermittlung der informatischen Grundkonzepte einteilen lässt, ausführlich vorgestellt.

Im empirischen Teil werden zu Beginn die konkreten Fragestellungen der Studie formuliert und die quantitativen und qualitativen Erhebungsinstrumente beschrieben (vgl. Kapitel 5). Die Eignung des Gesamtkonzepts zur Vermittlung der informatischen Grundkonzepte in Mathematik, Englisch, Deutsch und Musik in Klassenstufe 5 wird in Kapitel 6 zunächst aus der Sicht der beobachtenden Lehrerinnen und der betroffenen Schülerinnen und Schüler dargestellt. Die die Projektarbeit betreffenden Aspekte werden in Kapitel 6.2 behandelt. Darauf folgen die Ergebnisse aus dem Leistungstest zu den informatischen Grundkonzepten, der zu Beginn und am Ende des Schuljahrs durchgeführt wurde. Sie sind allgemein und getrennt nach Geschlecht und Fähigkeiten im sprachgebundenen, zahlengebundenen und formallogischen Denken ausgewertet. Zur Absicherung der gewonnenen Resultate werden einzelne Unterrichtsmaterialien wie Arbeitsblätter und Programme analysiert, die Aufschluss über das Erlernen der informatischen Grundkonzepte geben. Abschließend wird die Veränderung der Einstellungen zum Computer vom Beginn bis zum Ende des Schuljahrs diskutiert.

Ein zentrales Ergebnis der Forschungsarbeit zeigt, dass bereits Schülerinnen und Schüler der Klassenstufe 5 in einem kontinuierlichen Lernprozess über ein Schuljahr hinweg informatische Grundkonzepte im Fachunterricht eingebettet erlernen können. Dabei entstehen in der verkürzten Unterrichtszeit der Fächer keine erkennbaren fachlichen Defizite. Vielmehr werden allgemeine Fähigkeiten wie das Vorausdenken, das Strukturieren, das

Anwenden von Schemata, das genaue Arbeiten und Strategien zur Bewältigung von Frustrationen ausgebildet.

Das letzte Kapitel fasst die Arbeit zusammen und diskutiert die Erweiterung des Unterrichtskonzepts auf die Jahrgangsstufen 6 bis 10 und das für eine Umsetzung im Schulalltag der Realschulen notwendige Fortbildungskonzept für Lehrerinnen und Lehrer.

2 Informatische Bildung

Bereits in den sechziger Jahren wurden ganz vereinzelt in den Schulen alternative Formen des Lernens im programmierten Unterricht mit Computern erprobt (Schulmeister, 1996). Als die Hardware dann im Laufe der Zeit preisgünstiger und kleiner wurde und dadurch in immer mehr Schulen zur Verfügung stand (vgl. BMBF 2003), wurden verschiedene Konzepte in Bezug auf Informatiksysteme als Unterrichtsgegenstand und Werkzeug entwickelt. Diese erhielten Bezeichnungen wie *Informationstechnische Grundbildung (ITG)*, *Informationstechnische Bildung*, *Informatische Grundbildung*, *Informatische Bildung*, *Informatik-Grundbildung*, *Informatik*, *Medienerziehung*, *Informations- und Kommunikationstechnologie (IKT)* oder *Informations- und Kommunikationstechnik (IuK)*. Der Begriff der *Informatischen Bildung*, verstanden als

„[...] das Ergebnis von Lernprozessen, in denen Grundlagen, Methoden, Anwendungen, Arbeitsweisen und die gesellschaftliche Bedeutung von Informatiksystemen erschlossen werden.“ (GI, 2000, S. 1),

erweitert um die Medienerziehung, umfasst alle Konzepte und wird in dieser Arbeit als Oberbegriff verwendet.

Im Folgenden wird die historische Entwicklung der informatischen Bildung in Deutschland anhand verschiedener Empfehlungen bedeutender Organisationen und der Bildungspläne für die Realschulen und Gymnasien der Sekundarstufe I in Baden-Württemberg dargestellt. Im Anschluss daran werden die neuen *Bildungsstandards zur Informationstechnischen Grundbildung (ITG)* im Bildungsplan 2004 für Realschule (Kultusministerium Baden-Württemberg, 2004b) auf ihren allgemein bildenden Wert hin überprüft. Aus den dort geforderten und in der Fachdidaktik Informatik aktuell diskutierten Inhalten werden dann die in Klassenstufe 5 für ein echtes Fundament einer informatischen Bildung notwendigen Inhalte, die *informatischen Grundkonzepte*, abgeleitet.

2.1 Empfehlungen zu einer informatischen Bildung

Bereits 1963 gründete die IFIP¹ ein technisches Komitee für den Bereich der informatischen Ausbildung und innerhalb dieses 1966 eine Arbeitsgruppe *Informatics and ICT*² in

¹International Federation for Information Processing

²Information and Communication Technology

Secondary Education. In dieser wurden Informatiklehrpläne, die Rolle der Informatik in anderen Fachbereichen, der Einsatz von Informatiksystemen im Unterricht, der Einfluss der Informatik auf die Inhalte und Methoden des Lehrens und Lernens und Ideen zur Ausbildung von Lehrerinnen und Lehrern erarbeitet. In den 1970 bis 1972 entwickelten Empfehlungen (IFIP Technical Committee on Computer Education 1970, 1971 u. 1972) der Arbeitsgruppe stehen die digitale Informationsverarbeitung, die dafür notwendige Aufbereitung der Daten, Programmiersprachen, einige Hinweise zu verschiedenen Anwendungen, Algorithmen, Modellierung und die Informationsgewinnung im Mittelpunkt.

In Deutschland wurde 1976 die erste Empfehlung zum Informatikunterricht in der Sekundarstufe I von der Gesellschaft für Informatik (GI) veröffentlicht. In der darauf folgenden Zeit arbeiteten auch andere Organisationen wie die Bund-Länder-Kommission (BLK), die Gesellschaft der Mathematik (GDM) und der Deutsche Verein zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts (MNU) weitere Papiere aus. Abbildung 2.1 zeigt die wichtigsten in Deutschland verabschiedeten Empfehlungen, inhaltlich und zeitlich gruppiert. Zusammenfassungen der einzelnen Dokumente sind in Anhang A zu finden.

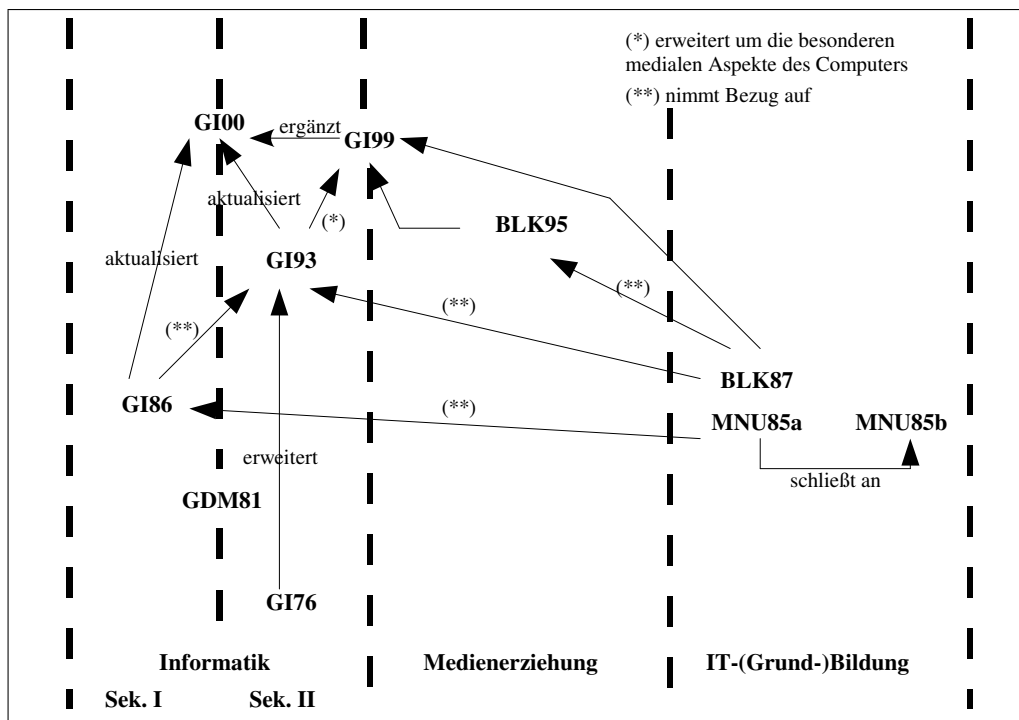


Abbildung 2.1: Entwicklung der Empfehlungen zur informatischen Bildung

Die Gesellschaft für Informatik, die 1969 mit dem Ziel, die Informatik zu fördern, gegründet wurde und im Moment ca. 24.000 Mitglieder aus allen Bereichen der Wissenschaft, der Informatikindustrie, der Anwendungen, der Lehre und der Ausbildung hat,

beeinflusste maßgeblich die Lehrplanentwicklung des Fachs Informatik mit zahlreichen Empfehlungen. Politisch einflussreicher war jedoch die Bund-Länder-Kommission, in der die Fragen des Bildungswesens, die sowohl den Bund als auch die Länder gemeinsam betreffen, diskutiert werden und die ihre Ergebnisse den Verantwortlichen auf Bundes- und Landesebene vorlegt. Außerdem entstanden in den achtziger Jahren vom Deutschen Verein zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts, einem der größten Fachlehrerverbände Deutschlands, und der Gesellschaft für Didaktik der Mathematik weitere Empfehlungen, in denen die informatische Bildung sehr stark mit Inhalten des Fachs Mathematik verknüpft wurden.

Die einzelnen Empfehlungen setzten ganz unterschiedliche Schwerpunkte. Sie beschäftigten sich mit Informatik, informationstechnischer (Grund-)Bildung und Medienerziehung. Die Trends lassen sich zeitlich unterscheiden und sind mit der Entwicklung der Informationstechnik gekoppelt. Bevor IBM 1981 den ersten Personal Computer (PC) auf den Markt brachte, wurden Informatiksysteme fast ausschließlich als Rechenmaschinen und zur Verwaltung großer Datenmengen eingesetzt. Die Bedienung war vor allem Spezialisten vorbehalten, die diese programmierten, um mit ihnen zu interagieren. So wurden auch in der ersten Empfehlung der GI (1976) *Zielsetzungen und Inhalte des Informatikunterrichts* und der GDM (1981) *Empfehlungen zur Einbeziehung informatischer Inhalte in den Mathematikunterricht der Sekundarstufe I Algorithmen und Programmiersprachen* als die zentralen Inhalte einer informatischen Bildung in den Vordergrund gestellt.

Nachdem der PC in den darauf folgenden Jahren immer stärker die Arbeitswelt durchdrang und in zunehmendem Maße in Privathaushalten zur Verfügung stand und durch die Entwicklung von Desktop-Publishing Software seine Bedienung erleichtert wurde, trat das Erlernen von Programmiersprachen zugunsten der *Bedienung von Anwendungssoftware* in den Hintergrund. Programmiersprachen sollten nur noch in dem Umfang eingeführt werden, dass bereits bestehende Programme verändert oder ergänzt werden konnten oder die für die Umsetzung einer algorithmischen Lösung in ein lauffähiges Programm notwendigen Sprachkonstrukte zur Verfügung standen. In der 1987 veröffentlichten Empfehlung *Gesamtkonzept für die informationstechnische Bildung – Informationstechnische Grundbildung* der BLK wurden Programmiersprachen im Zusammenhang mit Grundbildung überhaupt nicht mehr erwähnt. Dieser Trend wird dadurch verdeutlicht, dass nicht mehr von einem Fach Informatik, sondern von *Informationstechnischer Grundbildung* (ITG) (vgl. MNU 1985a und BLK 1987) oder *informationstechnischer Bildung* (vgl. MNU 1985b) gesprochen wurde. Beide besaßen einen viel stärkeren Anwendungsbezug und vernachlässigten die allgemeinen Grundlagen. Außerdem war für die Inhalte kein eigenes Fach mehr vorgesehen, sondern die informationstechnische Bildung wurde in Blockform in ein anderes Unterrichtsfach in Sekundarstufe I integriert. Zum ersten Mal wurden auch die individuellen Erfahrungen der Schülerinnen und Schüler im Umgang mit der Informationstechnik im Unterricht berücksichtigt (vgl. BLK 1987), da immer mehr im privaten Bereich Zugang zu Rechnern hatten und somit bereits mit Vorerfahrungen in den Unterricht kamen.

Dem Konzept der *Informationstechnischen Grundbildung* in Sekundarstufe I stellte die Gesellschaft für Informatik 1986 ihre *Rahmenempfehlung für die Informatik im Unterricht der Sekundarstufe I* (GI, 1986) gegenüber, die weiterhin die Umsetzung von *Algorithmen* in programmiersprachliche Formen und somit das Erlernen von Grundelementen höherer *Programmiersprachen* forderte. Die Erwartungen an ein Schulfach Informatik ging weiterhin über die reine Bedienung von Anwendungssoftware hinaus. Aufgrund der rasanten Entwicklung der Wissenschaft Informatik veröffentlichte die Gesellschaft für Informatik 1993 eine weitere Empfehlung für das Fach Informatik in der Sekundarstufe II (GI, 1993), die die von 1976 ersetzte. Inhaltlich stellte sie einen großen Einschnitt in die Entwicklung dar, da nicht mehr Programmiersprachen oder das Programmieren an sich, sondern *Entwurfsmethoden* im Vordergrund standen und die *Modellierung* in das Zentrum des Schulfachs Informatik gerückt wurde.

Bereits wenige Jahre nach der Einführung der *Informationstechnischen Grundbildung* zeigte sich, dass sie zum Scheitern verurteilt war. Sehr häufig gestalteten schlecht ausgebildete Lehrerinnen und Lehrer den Unterricht als Produktschulung und vermittelten keinerlei informatisches Grundlagenwissen. Die aus diesem Missstand resultierende Kritik führte zu einer erneuten Aufwertung der informatischen Inhalte. Parallel dazu wurden *mediale Aspekte des Computers* im Rahmen einer Medienerziehung stärker berücksichtigt. Die Empfehlungen *Medienerziehung in der Schule* der BLK (1995) und *Informatische Bildung und Medienerziehung* der GI (1999) beschrieben die Erwartungen an eine Medienerziehung. Letztere stellte die *informatischen Inhalte, Methoden und Sichtweisen*, die erst ein Verständnis des Mediums Computer ermöglichen, und nicht die Nutzung des Rechners in den Mittelpunkt.

In der neuesten *Empfehlung für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen* der GI (2000) wurde zum ersten Mal der Einstieg in die Informatik in der *Primarstufe* beschrieben. Ein Computersystem soll dort funktional zur *Lösung von konkreten Aufgaben* eingesetzt werden. Für die Sekundarstufe I forderte die GI dann ein eigenständiges Fach, das möglichst früh in den Pflichtkanon der Fächer aufgenommen wird, um „rechtzeitig Fach-, Methoden-, Sozial- und Selbstkompetenz im Umgang mit Information, insbesondere digital dargestellter, sowie mit Informatiksystemen auszuprägen.“ (GI, 2000, S. 5). Insgesamt baute das Dokument sehr stark auf der von der Gesellschaft für Informatik 1993 veröffentlichten Empfehlung auf. Lediglich der Begriff der *Information* erhielt in der Leitlinie *Interaktion mit Informatiksystemen* eine neue zentrale Rolle, indem die Wichtigkeit der Strategien im Umgang mit Informationen betont wurden.

Die inhaltliche Entwicklung der Empfehlungen in Deutschland verlief also von der Konzentration auf Programmiersprachen und Algorithmen über die Hinwendung zu einem sehr starken Anwendungsbezug der Inhalte in der *Informationstechnischen Grundbildung* und nach ihrem Scheitern zu einer Rückbesinnung auf die allen Anwendungssystemen zugrunde liegenden informatischen Konzepte.

2.2 Bildungspläne der Sekundarstufe I in Baden-Württemberg

Die im vorherigen Kapitel skizzierte Entwicklung der Empfehlungen findet sich zeitlich versetzt in den Rahmenlehrplänen der einzelnen Bundesländer wieder. Stritzky (1995) hat zum Beispiel die inhaltlichen Schwerpunkte des ITG-Unterrichts in den alten Bundesländern bezüglich der in der Empfehlung *Gesamtkonzept für die informationstechnische Bildung* (BLK, 1987) vorgeschlagenen Aufgaben der informationstechnischen Bildung untersucht. Bis auf die Entwicklung der elektronischen Datenverarbeitung, die nur im Unterricht in zwei Bundesländern thematisiert wird, werden die anderen Aufgaben in fast allen Rahmenlehrplänen berücksichtigt.

In diesem Kapitel wird zunächst die historische Entwicklung der in den Lehrplänen für die Realschule und das Gymnasium in Baden-Württemberg ausgewiesenen Lehrplaneinheiten und Unterrichtsfächer, die zu einer informatischen Bildung beitragen, vorgestellt. Außerdem werden ihre Bezüge zu den Empfehlungen aufgezeigt. Die Lehrpläne der Hauptschule finden keine Berücksichtigung, da sie nur in sehr geringem Umfang informatische Inhalte fordern. Speziell die Inhalte der *Bildungsstandards für Informationstechnische Grundbildung* im Bildungsplan 2004 der Realschule werden genauer auf ihren allgemein bildenden Charakter und hinsichtlich der Berücksichtigung der in den Empfehlungen vorgeschlagenen Themen analysiert. Daraus werden in Kapitel 2.4 die für eine informatische Bildung in Klasse 5 der Realschule notwendigen informatischen Grundkonzepte abgeleitet.

2.2.1 Historische Entwicklung

Seit 1984 werden in Baden-Württemberg im Rhythmus von zehn Jahren vom Kultusministerium für Wissenschaft, Bildung und Forschung neue Bildungspläne herausgegeben, die die geänderten Anforderungen an eine schulische Bildung berücksichtigen. In dieser Arbeit werden die Teile der Bildungspläne für das Gymnasium und die Realschule analysiert, die einen Beitrag zur informatischen Bildung leisten.

Die Bildungspläne von 1984 zeichnen das Fach Mathematik aus, Grundkenntnisse über Computer und Informatik zu vermitteln. Sie enthalten jeweils in Klassenstufe 9 eine Lehrplaneinheit. In der Realschule geht es um die Formulierung von Lösungsabläufen für einfache Probleme, die auf dem Computer ausgeführt werden können. Im Bildungsplan für das Gymnasium wird der ganze Prozess der Erstellung einer Problemlösung mit Hilfe des Computers betont und die Angemessenheit des Einsatzes von Computern thematisiert (siehe Tabelle 2.1). Beide Bildungspläne orientieren sich sehr stark an der 1981 von der GDM verabschiedeten *Empfehlung zur Einbeziehung informatischer Inhalte in den Mathematikunterricht der Sekundarstufe I* (GDM, 1981). Die gesellschaftlichen Auswirkungen der Datenverarbeitung und die theoretischen und technischen Grundlagen

der Informatik, die in der ersten GI-Empfehlung 1976 für die Sekundarstufe II formuliert sind, werden hier nicht berücksichtigt.

1984	Realschule – Klasse 9
	LPE „Sachrechnen und Computer“ (Mathematik)
	* Programmierung
1984	Gymnasium – Klasse 9
	LPE „Elemente der Informatik“ (Mathematik)
	* Analyse eines Problems, Erstellung eines Algorithmus, Programmierung, Test der Lösung und Dokumentation * Beurteilung der Einsatzmöglichkeit eines Computers

Tabelle 2.1: Grundkenntnisse über Computer und Informatik in den Bildungsplänen von 1984 (Kultusministerium Baden-Württemberg 1984b u. 1984a)

In den Bildungsplänen von 1994 (vgl. Tabelle 2.2) spiegeln sich dann deutlich die Empfehlungen der MNU (1985a und 1985b) und der BLK (1987) wider.

1994	Realschule – Klasse 7
	<i>Informationstechnische Grundbildung</i> (eigenes Fach)
	* Einweisung in den Computerarbeitsplatz * Einführung in die Textverarbeitung * Erste Erfahrungen mit dem Computer
	Realschule – Klasse 8
	<i>Informationstechnische Grundbildung</i> (eigenes Fach)
	* Informationstechnische Werkzeuge * Einsatzmöglichkeiten * Auswirkungen
1994	Gymnasium – Klasse 8
	LPE „ <i>Informationstechnische Grundkenntnisse</i> “ (Mathematik) Bis Klasse 11 soll der Computer in verschiedenen Fächern praxisbezogen als Werkzeug und Medium im Rahmen der Informationstechnischen Grundbildung eingesetzt werden
	* Informationstechnische Werkzeuge * Einsatzmöglichkeiten * Erstellung und Ausführung einfacher Programme * Auswirkungen

Tabelle 2.2: Lehrpläne *Informationstechnische Grundbildung* und *Informationstechnische Grundkenntnisse* von 1994 (Kultusministerium Baden-Württemberg 1994b u. 1994a)

Die informatischen Inhalte werden zugunsten der Anwendungen zurückgedrängt. Dieser Aspekt wird noch dadurch verstärkt, dass zum Beispiel die in der Empfehlung der MNU formulierten informatischen Themen wie die Anpassung von Anwendungen, die Modellierung oder die Problemlösung mit dem Computer keine oder kaum Berücksichtigung finden. Neu in Bezug auf die Bildungspläne von 1984 ist die explizite Beschäftigung mit den gesellschaftlichen Auswirkungen der Informationstechnik.

In der Realschule ist zum ersten Mal ein eigenes Fach über zwei Schuljahre hinweg ausgewiesen. Dadurch steht genügend Zeit zur Verfügung, eine solide Grundbildung der festgelegten Inhalte zu vermitteln. Im Bildungsplan des Gymnasiums wird dagegen nur eine Lehrplaneinheit im Fach Mathematik formuliert. Eine Integration des Computers als Werkzeug und Medium in den Unterricht anderer Fächer bleibt engagierten Lehrerinnen und Lehrern überlassen. Findet sie vor der 7. bzw. 8. Klassenstufe statt, muss die Einarbeitung extra erfolgen.

In den neuen Bildungsplänen 2004 ist auch in der Realschule wieder auf ein eigenes Fach verzichtet worden. In beiden Schulformen soll eine *Informationstechnische Grundbildung* in allen Klassenstufen der Sekundarstufe I integriert vermittelt werden (siehe Tabelle 2.3).

2004	Realschule – Klassen 5 - 10
	Informationstechnische Grundbildung, integriert in nicht näher spezifizierte Fächer
	<ul style="list-style-type: none"> * Arbeiten und Lernen mit informationstechnischen Werkzeugen * Zusammenarbeiten und Kommunizieren * Entwickeln, Zusammenhänge verstehen, Reflektieren
2004	Gymnasium – Klassen 5 - 10
	Informationstechnische Grundbildung, integriert in nicht näher spezifizierte Fächer
	<ul style="list-style-type: none"> * Selbstständiges Arbeiten und Lernen mit informationstechnischen Werkzeugen * Erfolgreich Zusammenarbeiten und Kommunizieren * Entwickeln, Zusammenhänge verstehen und Reflektieren

Tabelle 2.3: *Bildungsstandards für Informationstechnische Grundbildung* für die Sekundarstufe I an Realschulen und Gymnasien von 2004 (Kultusministerium Baden-Württemberg 2004b u. 2004a)

Im Gymnasialbereich ist projektorientiertes und in der Realschule produktorientiertes Arbeiten vorgesehen. Die Arbeit soll die Medienkompetenz der Schülerinnen und Schüler fördern. Die drei Kompetenzbereiche *Arbeiten und Lernen mit informationstechnischen Werkzeugen*, *Zusammenarbeiten und Kommunizieren* und *Entwickeln, Zusammenhänge verstehen, Reflektieren* stimmen in beiden Schularten überein. Die höheren Anforderun-

gen im gymnasialen Bereich werden durch die qualitativen Beschreibungen für die ersten beiden Bereiche zum Ausdruck gebracht. Die Inhalte beider Bildungspläne können den in der Empfehlung der GI (1999) herausgestellten medialen Funktionalitäten des Computers zugeordnet werden. Das *Arbeiten und Lernen mit informationstechnischen Werkzeugen* entspricht der Interaktion, das *Zusammenarbeiten und Kommunizieren* der Vernetzung und das *Entwickeln, Zusammenhänge verstehen und Reflektieren* zum Teil dem Verarbeiten von Daten. Die konkreten Inhalte sind in den Bildungsplänen oberflächlicher als in der Empfehlung der Gesellschaft für Informatik formuliert. Keiner der Kompetenzbereiche berücksichtigt zum Beispiel die für eine informatische Bildung sehr wichtigen Inhalte der Modellierung und der Verfahren der Softwareentwicklung. Tiefer gehende informatische Sichtweisen und Methoden, die in der Empfehlung *Informatische Bildung und Medienerziehung* (GI, 1999) gefordert werden, spielen keine Rolle.

2.2.2 Bildungsstandards für Informationstechnische Grundbildung an Realschulen

Die für eine *Informationstechnische Grundbildung* notwendigen Kompetenzen werden in den Bildungsstandards, wie weiter oben beschrieben, in drei Bereiche aufgeteilt. Ihnen sind in Abbildung 2.2 die inhaltlich in Gruppen zusammengefassten Kompetenzen und Inhalte zugeordnet. Sie bilden ein Fundament, aus dem nach den Bildungsstandards in systematischen Lernprozessen allgemeine Fähigkeiten wie das Abstraktionsvermögen, die Modellierung etc. entwickelt werden können (Kultusministerium Baden-Württemberg, 2004b, S. 189). Die Fähigkeiten sind in Wolken eingebettet, da aus den Bildungsstandards nicht hervorgeht, unter welchen konkreten Bedingungen sie gefördert werden.

Der Umgang mit Soft- und Hardware steht in allen drei Bereichen sehr stark im Vordergrund. Außerdem nimmt die Thematisierung der gesellschaftlichen Auswirkungen eine zentrale Rolle ein. Grundlegenden Aspekten wird dagegen sehr wenig Platz eingeräumt. Beim *Zusammenarbeiten und Kommunizieren* sind es nur die Beschreibung der Organisationsstruktur vernetzter Umgebungen und im Bereich *Entwickeln, Zusammenhänge verstehen, Reflektieren* die Darstellung des Aufbaus von Datenverarbeitungssystemen und der Umgang mit Grundbegriffen der Digitalisierung. Dies widerspricht der in der Empfehlung *Informatische Bildung und Medienerziehung* formulierten inhaltlichen Forderung an eine informatische Bildung:

„Strukturen, die für bestimmte Abläufe erforderlich sind, oder Arbeitsweisen des Systems, die bestimmte Funktionalitäten ermöglichen, werden analysiert und auf grundlegende verallgemeinerbare Prinzipien zurückgeführt. Die informationstechnologischen Zusammenhänge werden auf diese Weise durchschaubarer und für eigene Gestaltungsaufgaben verfügbar gemacht.“ (GI, 1999)

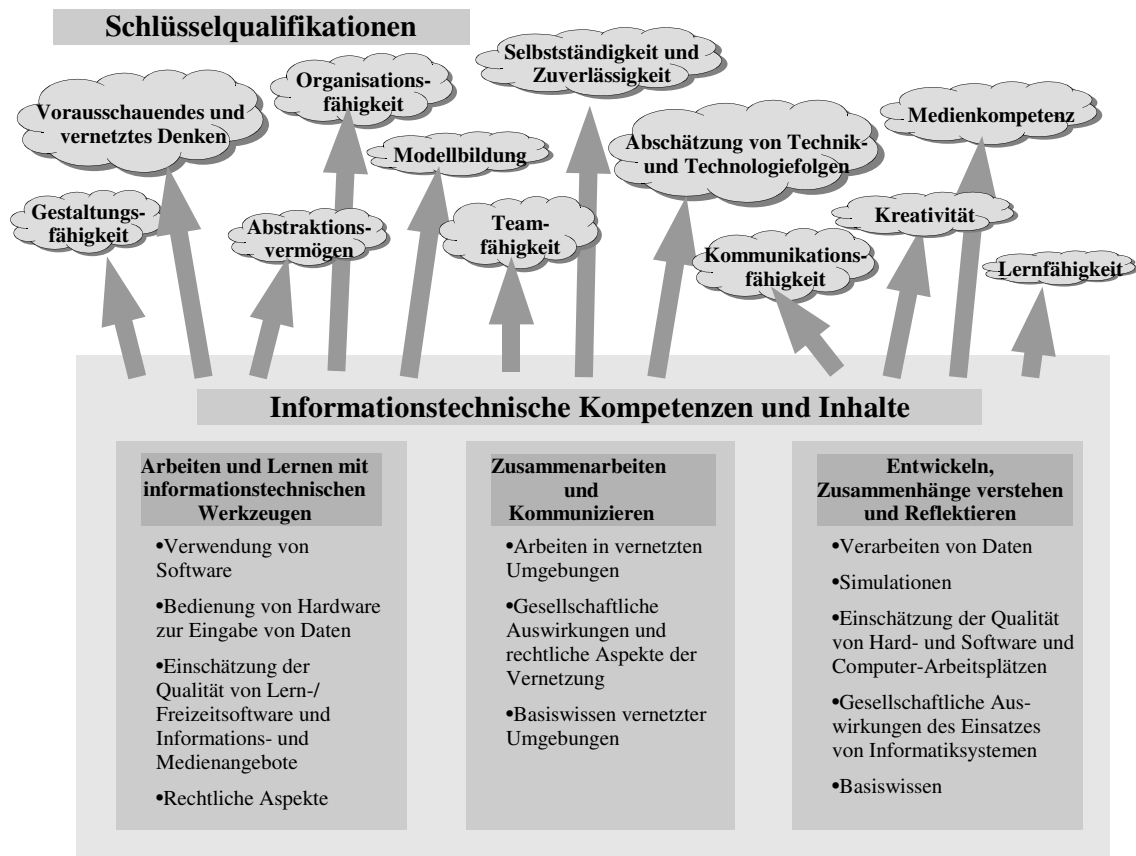


Abbildung 2.2: *Bildungsstandards für Informationstechnische Grundbildung an Realschulen: Kompetenzen und Inhalte* (Kultusministerium Baden-Württemberg, 2004b)

Auch den Forderungen von Lötke (2004), die Lernenden dahingehend auszubilden, dass sie nicht nur die gerade anstehenden Probleme bewältigen können, sondern den ständig neu auftretenden Anforderungen gewachsen bleiben, wird durch die Bildungsstandards nicht Rechnung getragen.

2.3 Standpunkte der Fachdidaktik Informatik

Die Vertreter der Fachdidaktik Informatik im deutschsprachigen Raum präsentieren ganz unterschiedliche inhaltliche Vorschläge für eine informatische Bildung. Schubert und Schwill (2004) suchen nach einem Fundamentum als Quelle für Gegenstände, die im Informatikunterricht vermittelt werden sollen. Dieses muss ihrer Meinung nach existieren, da es trotz der verschiedenen Paradigmenwechsel im Bereich der Informatik keine wissenschaftlichen Revolutionen, sondern nur Weiterentwicklungen gegeben hat. Das

Fundamentum sichert „eine gewisse Aktualität bei gleichzeitiger Beständigkeit des Informatikunterrichts“ (Schubert und Schwill, 2004, S. 73). Seine Inhalte werden aus den beim Software-Entwicklungsprozess typischen Vorgehensweisen von Informatikern abgeleitet und müssen den Kriterien einer fundamentalen Idee genügen. Schubert und Schwill (2004, S. 85) definieren eine *fundamentale Idee* folgendermaßen:

„Eine fundamentale Idee bezgl. eines Gegenstandsbereichs (Wissenschaft, Teilgebiet) ist ein Denk-, Handlungs-, Beschreibungs- oder Erklärungsschema, das

1. in verschiedenen Gebieten des Bereichs vielfältig anwendbar oder erkennbar ist (*Horizontalkriterium*),
2. auf jedem intellektuellen Niveau aufgezeigt und vermittelt werden kann (*Vertikalkriterium*),
3. zur Annäherung an eine gewisse idealisierte Zielvorstellung dient, die jedoch faktisch möglicherweise unerreichbar ist (*Zielkriterium*),
4. in der historischen Entwicklung des Bereichs deutlich wahrnehmbar ist und längerfristig relevant bleibt (*Zeitkriterium*) und
5. einen Bezug zu Sprache und Denken des Alltags und der Lebenswelt besitzt (*Sinnkriterium*).“

Auf Basis der Analyse der Phasen des Software-Life-Cycles (*Problemanalyse*, *Entwurf*, *Implementierung*, *Funktionsüberprüfung* und *Leistungsüberprüfung*) werden die drei Masterideen *Algorithmisierung*, *Sprache* und *strukturierte Zerlegung* identifiziert. Abbildung 2.3 zeigt diese mit den ihnen untergeordneten Ideen.

Hubwieser (2001) leitet die für ein Fach Informatik relevanten Inhalte von dem zentralen Begriff der *Information* ab und spricht von einem *informationszentrierten Ansatz*. Ausgehend von dem Grundschema jeder künstlichen Informationsverarbeitung, identifiziert er die *Repräsentation von Informationen mit verschiedenen Darstellungstechniken*, die *Verarbeitung und den Transport dieser Repräsentation*, aus der dadurch eine neue Repräsentation entsteht, und zuletzt die *Interpretation der neuen Repräsentation*. Die Masterideen *Algorithmisierung*, *Sprache* und *strukturierte Zerlegung* von Schubert und Schwill (2004) sieht er in den Bereichen Verarbeitung, Darstellung und Verteilung berücksichtigt. Für die Auswahl konkreter informatischer Inhalte fasst er das Horizontal- und Sinnkriterium zu einem Kriterium größtmöglicher Allgemeingültigkeit zusammen und übernimmt das Zeit- und Vertikalkriterium. Außerdem führt er als neue Anforderung die exemplarische Auswahl von Inhalten und die Möglichkeit zu ihrer Einflechtung in andere Themen ein. Zentral ist für ihn die Modellierung. Sie erzeugt

„[...] eine Hilfebene zwischen der Problem- und der Implementierungsebene, die es erlaubt, im Lehrplan alle wesentlichen Anforderungen an den Unter-

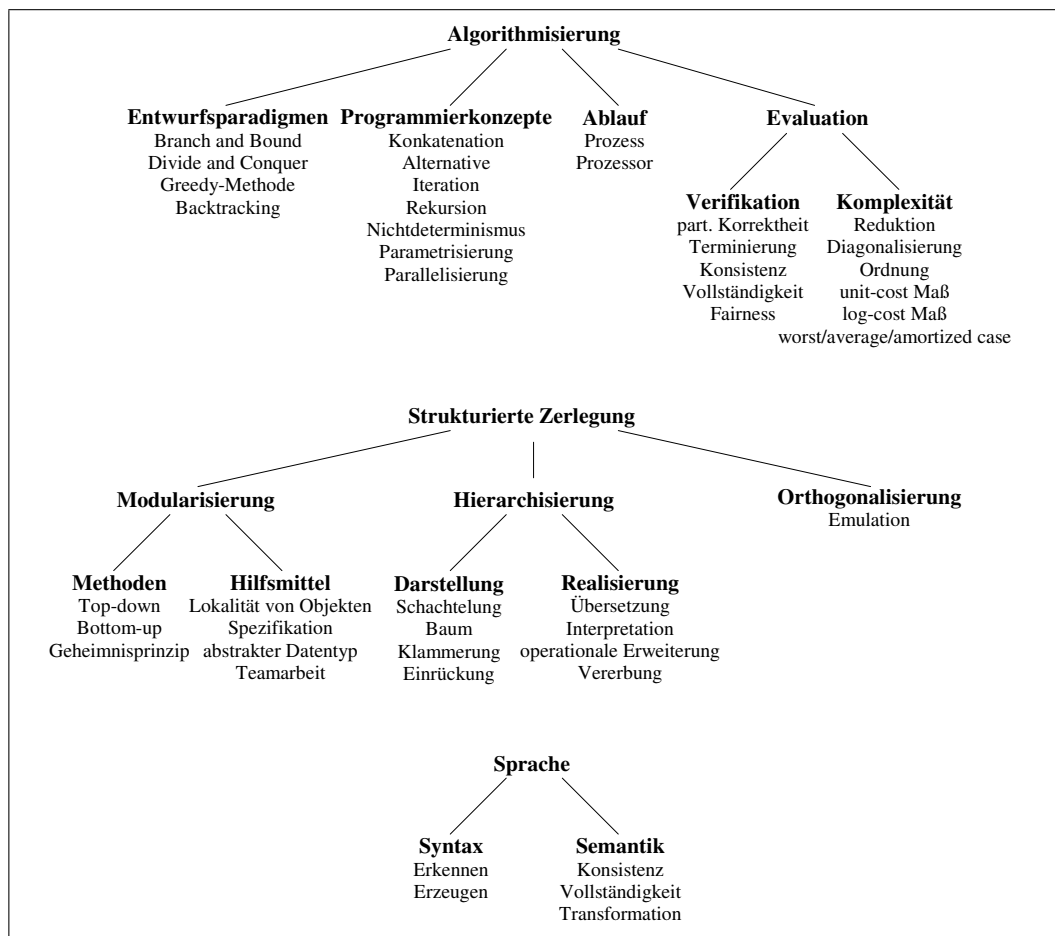


Abbildung 2.3: Fundamentale Ideen der Informatik (Schubert und Schwill, 2004, S. 97)

richt zu formulieren, ohne auf programmiertechnische Details wie spezielle Datenstrukturen oder bestimmte Algorithmen eingehen zu müssen.“ (Hubwieser, 2001, S. 90)

Anhand verschiedener Modellierungstechniken sollen Modelle entwickelt und möglichst implementiert werden. Im Vordergrund steht immer die Konstruktion der Modelle und nicht ihre Implementierung in einer bestimmten Programmiersprache.

Hartmann und Nievergelt (2002) bezeichnen die Informatik als Basiswissenschaft. Unterricht muss deshalb die folgenden drei Ziele verfolgen:

- Vermittlung der für den Alltag unabdingbaren Fertigkeiten
- Einblick in das Wesen und die intellektuellen Errungenschaften des Fachs
- Schaffung eines Bewusstseins für die Bedeutung und Rolle des Fachs in einem größeren gesellschaftlichen Rahmen.

Um diese Ziele zu erreichen, müssen verschiedene Facetten der Informatik betrachtet werden. Der *Informatik-Turm* (Nievergelt, 1995) in Abbildung 2.4 stellt die verschiedenen Aspekte dar. Die Schichten des Bauwerks sind zwar unterschiedlich gewichtet, der Turm würde ohne Fundament jedoch zusammenbrechen. Für eine fundierte, allgemein bildende und trotzdem praxisorientierte Ausbildung in der Basiswissenschaft Informatik wird deshalb von Reichert & al. (2004, S. 4) gefordert, dass der *Informatik-Turm* in seiner ganzen Höhe zumindest besichtigt wird. Dadurch wird es erst möglich, dass komplexe Informatiksysteme sinnvoll und angemessen eingesetzt werden.

“Der Computerbenutzer muss die technische Spezifikation des unsichtbaren Eisbergs nicht kennen. Aber er sollte wissen, dass sich unter der Oberfläche eine gewaltige Infrastruktur versteckt und warum die sichtbare Spitze überhaupt schwimmt.“ (Reichert & al., 2004, S. 5)

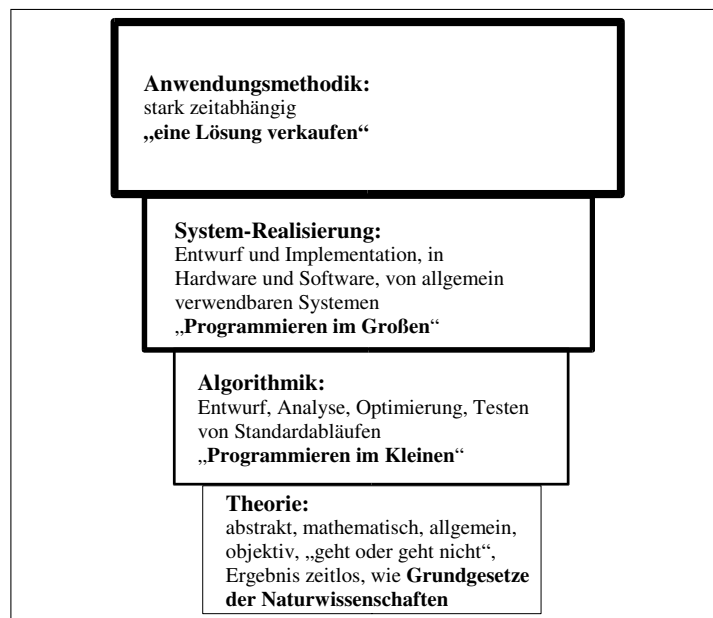


Abbildung 2.4: *Informatik-Turm* (Nievergelt, 1995)

Wie Nievergelt polarisiert auch Friedrich (2003) nicht zwischen der Benutzung von Informatiksystemen und der Vermittlung ihrer theoretischen Grundlagen in einem Schulfach Informatik. Als inhaltlichen Ausgangspunkt für die Entwicklung seines Stufenmodells einer informatischen Bildung wählt er die von der Gesellschaft für Informatik formulierten Leitlinien informatischer Bildung (GI, 2000).

“Letztlich ist es doch nur auf der Basis grundlegender Kenntnisse zu Interaktionen, Wirkprinzipien und Modellen von Informatiksystemen erst sinnvoll, über Möglichkeiten und Grenzen von Computern zu diskutieren und deren gesellschaftliche Wirkung im Informationszeitalter einschätzen zu lernen.“ (Friedrich, 2003, S. 134)

In Anlehnung an die Kompetenzen der mathematischen oder naturwissenschaftlichen Grundbildung der PISA-Studie (Deutsches PISA-Konsortium, 2001) formuliert er fünf informatische Kompetenzstufen über alle Leitlinien informatischer Bildung hinweg und differenziert sie wie in Tabelle 2.4 dargestellt.

	Leitlinien informatischer Bildung			
	Interaktion mit Informatiksystemen (IS)	Wirkprinzipien von IS	Informatische Modellierung	Wechselwirkung IS, Mensch und Gesellschaft
Stufe I	Bedienung von Informatikanwendungen			
	Einfache Bedienung; Nachvollzug von Handlungen	Benennung von Teilen eines Computerarbeitsplatzes	Erfassen typischer Bestandteile und Eigenschaften	Beziehungen in der Umgangssprache darstellen
Stufe II	Benutzung von Informatiksystemen			
	Komplexe Bedienung; Erfassen von Abläufen	Wissen um Grundfunktionen von IS	Modellhafte Darstellung von Abläufen	Computer in unterschiedlichen Lebenssituationen
Stufe III	Kenntnis fachsystematischer Grundlagen			
	Fertigkeiten zum Lösen typischer Aufgabenklassen	Einordnung in Fachsystematik; grundlegende Fachbegriffe	Definition und Anwendung des Modellbegriffs	Kritische Reflexion zur Nutzung von Informationen; historische Entwicklung
Stufe IV	Verständnis von Konzepten der Informatik			
	Umgang mit Systemen; Auswahl von Methoden	Theoretische Grundlagen; Fachbegriffe und Konzepte	Einfache Modelle entwickeln und implementieren	Beurteilung von Auswirkungen
Stufe V	Entwicklung und Bewertung von Informatiksystemen			
	Konstruktion und Implementierung von Lösungen mit verschiedenen Werkzeugen	Anwenden und Entwickeln von Konzepten	Komplexere Modelle mit unterschiedlichen Werkzeugen bearbeiten	Bewertung des Einsatzes von Informatiksystemen

Tabelle 2.4: Zusammenhang der Leitlinien informatischer Bildung und informatischer Kompetenz (nach Friedrich, 2003, S. 135 und S. 140)

Auch Puhlmann (2003) greift auf die PISA-Studie und die Empfehlungen der GI (2000) zurück. Orientiert an dem allgemeinen Grundbildungsbegriff von PISA, definiert er zunächst *informatische Literalität*:

“Informatische Literalität ist die Fähigkeit einer Person, die Rolle zu erkennen und zu verstehen, die Informatik und Informatiksysteme in der Welt spielen, fundierte auf informatischem Wissen beruhende Urteile abzugeben und sich auf eine Weise mit der Informatik und ihren Anwendungen zu befassen, die den Anforderungen des gegenwärtigen und künftigen Lebens dieser Person als konstruktivem, engagiertem und reflektierendem Bürger entspricht.“ (Puhlmann, 2003, S. 148)

Welche Informatikinhalt Basis informatischer Literalität sind, leitet er aus den informatischen Tätigkeiten der Jugendlichen in ihrer Lebenswirklichkeit ab. Er identifiziert die drei Kompetenzklassen *Anwendung*, *Gestaltung* und *Entscheidung* und ordnet ihnen Tätigkeiten und informatisches Wissen zu (siehe Tabelle 2.5).

Anwendung	
Tätigkeit:	Anwendung von Informatiksystemen zur Lösung von Problemen aus dem persönlichen Umfeld
Gestaltung	
Tätigkeit:	Gestaltung von Informatiksystemen auf Grundlage von Modellierungsprozessen; häufig unter Verwendung bereits bestehender Informatiksysteme
Wissen:	Verständnis der Funktionsweise und Struktur der eingesetzten Informatiksysteme Allgemeines informatisches Fachwissen zur Erstellung eines informatischen Modells
Entscheidung	
Tätigkeit:	Verantwortungsvolles Entscheiden über den Einsatz und die Entwicklung von Informatiksystemen; reflektierte Einschätzung der Auswirkungen von Informatiksystemen im gesellschaftlichen Kontext sowie auf das Individuum
Wissen:	Prinzipielle und praktische Möglichkeiten und Grenzen von Informatiksystemen

Tabelle 2.5: Kompetenzklassen nach Puhlmann (2003, S. 148)

(Wedekind & al., 2004a) fordert eine Informatik als Grundbildung. Dazu gehört seiner Meinung nach die Vermittlung von Basiskompetenzen, die Voraussetzung für den Erwerb einer Allgemeinbildung sind. Ohne Anspruch auf Vollständigkeit benennt er folgende sechs Themen, die dabei eine entscheidende Rolle spielen:

- *Schema und Ausprägung* verstanden als universelle und spezielle, singuläre Beschreibung von Gegenständen;
Schemata sind situationsunabhängig und beliebig oft wiederholbar. In der Informatik sind Klassen, Relationen oder Programmcode Beispiele für Schemata und

Objekte, Tupel oder ablaufende Programme Beispiele für Ausprägungen. Insgesamt ist der Anteil der computerisierbaren Schemata gering, aber sie bedeuten „heute im Alltagsleben bei der täglichen Problembewältigung sehr viel“ (Wedekind & al., 2004a).

- *Bildung von Elementarsätzen* nach einer rationalen Grammatik, mit der alle Sprachen beschrieben werden können;
Eine rationale Grammatik ist die Grundlage aller Sprachen, die der Rechner versteht. Aufgrund ihres Anspruchs auf Allgemeingültigkeit für alle Sprachen muss sie sehr einfach sein. Dadurch ist sie für die menschliche Kommunikation zu primitiv, jedoch kann mit ihrer Hilfe eine „methodische Rekonstruktion unserer gesprochenen Sprache“ (Wedekind & al., 2004b, S. 265) vorgenommen werden. Durch die Schemabildung bei der Rekonstruktion ist eine Automatisierung von Sprache erst möglich.
- *Gleichheit und Abstraktion*
Im Prozess der Abstraktion wird in einer Menge verschiedener Objekte nach Eigenschaften gesucht, die in gewisser Hinsicht gleich sind. Die Gleichheit wird über Äquivalenzrelationen beschrieben, die konkret benannt werden. Im Prinzip ist in der Informatik jede Schnittstelle eine Abstraktion (Wedekind & al., 2004c).
- *Objektsprache/Metasprache*
Notwendige Voraussetzung für Kommunikation ist das Verständnis einer gemeinsamen Sprache, d.h. eines gemeinsamen Schemas. Dieses muss allerdings selbst wieder mit Hilfe einer Sprache, einer Schemabeschreibungssprache, beschrieben werden. Dabei handelt es sich um eine Metasprache, in der die Grammatik der Kommunikation festgelegt ist (Wedekind & al., 2004d).
- *Namensgebung und Kennzeichnung*
Um mit Objekten zu arbeiten, müssen sie referenziert werden können. Wedekind & al. (2004a) behaupten, dass die reale Welt nahezu namenlos ist und aus diesem Grund die Dinge gekennzeichnet werden müssen. Insbesondere gilt dies natürlich für virtuelle Objekte in Informatiksystemen, so dass Kennzeichnungen in der Informatik häufig auftreten.
- *Logik und Geltungssicherung von Behauptungen*
Zur Entwicklung von Strukturen in jedem „Durcheinander“ müssen Ordnungsregeln angewendet werden. Die dialogische Logik bietet nach Meinung von Wedekind & al. (2005) mögliche Hilfsmittel zur Strukturierung.

Tabelle 2.6 fasst die verschiedenen Positionen der Fachdidaktik Informatik noch einmal zusammen.

Bezeichnung	Ausgangspunkt	Schwerpunkte	Bedingungen
Schubert und Schwill (2004)			
Fundamentale Ideen der Informatik	Softwareentwicklungsprozess	Algorithmisierung, Sprache, strukturierte Zerlegung	Horizontal-, Vertikal-, Ziel-, Zeit- und Sinnkriterium
Hubwieser (2001)			
Informationszentrierter Ansatz	Grundschema jeder künstlichen Informationsverarbeitung	Darstellung von Information, Verarbeitung und Transport von Repräsentation, Interpretation von Repräsentation	Allgemeine Bedeutung, Lebensdauer, Vermittelbarkeit, exemplarische Auswahl/Verflechtung
Reichert & al. (2004), Nievergelt (1995)			
Informatik-Turm	Konzepte jeder Informatiklösung	Anwendungsmethodik, System-Realisierung, Algorithmik und Theorie	Langlebige, verallgemeinerungsfähige Begriffe und Prinzipien
Friedrich (2003)			
Informatische Kompetenz	Leitlinien aus GI (2000), Kompetenzstufen der PISA-Studie	Bedienung, Benutzung von IS, Kenntnis fachsystematischer Grundlagen, Verständnis von Konzepten der Informatik, Entwicklung und Bewertung von IS	
Puhlmann (2003)			
Informatische Literalität	Grundbildungsbegriff der PISA-Studie, GI (2000)	Anwendung, Gestaltung, Entscheidung	Inform. Tätigkeiten der Jugendlichen in ihrer Lebenswirklichkeit in einer Industrienation
Wedekind & al. (2004a)			
Informatik als Grundbildung, informatical literacy	Informationsverarbeitung	Schema und Ausprägung, Bildung von Elementarsätzen, Gleichheit und Abstraktion, Objektsprache/ Metasprache, Namensgebung und Kennzeichnung, Logik und Geltungssicherung von Behauptungen	Basiskompetenzen

Tabelle 2.6: Positionen der Fachdidaktik Informatik

2.4 Informatische Grundkonzepte in Klasse 5 der Realschule

Im Bildungsplan 2004 für Realschulen in Baden-Württemberg wird im Erziehungs- und Bildungsauftrag gefordert, dass die Schule „auf lebenslanges und nachhaltiges Lernen, auf praktische Lebensbewältigung in Alltag und Beruf sowie auf aktive Teilhabe am gesellschaftlichen Leben“ vorbereitet und eine „allgemeine Bildung“ (Kultusministerium Baden-Württemberg, 2004b, S. 5) vermittelt. Speziell für die Informationstechnische Grundbildung wird folgende Bedeutung herausgestellt:

“Die Informationstechnische Grundbildung (ITG) ist ein wesentlicher Bestandteil zeitgemäßer Allgemeinbildung. Wichtig ist dabei die Fähigkeit, Informations- und Kommunikationstechniken kritisch zu beurteilen sowie in individueller und sozialer Verantwortung zu nutzen.“ (Kultusministerium Baden-Württemberg, 2004b, S. 6)

Durch die zu starke Hervorhebung der Anwendung von Informatiksystemen und die Vernachlässigung von Grundlagenwissen werden die *Bildungsstandards für Informationstechnische Grundbildung* 2004 in Baden-Württemberg den Ansprüchen einer Allgemeinbildung und der Unterstützung von lebenslangem Lernen nicht gerecht. Für den Einstieg in eine Grundbildung in Klassenstufe 5 erscheinen nach Ansicht der Autorin dieser Arbeit folgende Inhalte relevant:

- **Wirkprinzipien von Informatiksystemen:** Algorithmisierung, Automatisierung und Codierung
- **Informatische Modellierung:** Datenstrukturen, Modellierung und Objektorientierung
- **Arbeitstechniken:** Modularisierung, Hierarchisierung, Abstraktion, Iteration und Rekursion
- **Informatische Beschreibungsmittel:** Programmablaufpläne, Syntaxdiagramme, UML-Klassendiagramme, Hierarchien und Netzwerke
- **Fachnahe Inhalte:** Grammatik, Syntax, Semantik und Funktionsbegriff.

Diese werden im Folgenden als **informatische Grundkonzepte** bezeichnet. Ihre Auswahl wird von der Autorin durch die Verifizierung der Kriterien fundamentaler Ideen, wie sie bereits in Wursthorn (2005c) skizziert wurden, begründet. Die Standpunkte der Fachdidaktik Informatik bilden die Basis für das Horizontalkriterium, da dort über ganz verschiedene Bereiche der Informatik diskutiert wird. Das Zeitkriterium und damit die längerfristige Relevanz der informatischen Grundkonzepte werden durch die Analyse der Entwicklung der Empfehlungen zur informatischen Bildung abgeleitet. Dem Zielkriterium werden die in den *Bildungsstandards für Informationstechnische Grundbildung* (Kultusministerium Baden-Württemberg, 2004b) genannten Schlüsselqualifika-

tionen, wie die Fähigkeit zum Abschätzen von Technik- und Technologiefolgen, vorausschauendes und vernetztes Denken, Modellierung, Abstraktionsvermögen, Kreativität, Selbstständigkeit und Zuverlässigkeit, Organisations-, Kommunikations-, Team-, Gestaltungs- und Lernfähigkeit, zugeordnet. Das Sinnkriterium wird aus der Perspektive der Anwendung von Informatiksystemen in der Schule, im späteren Berufsleben und in der Freizeit betrachtet. Allein das Vertikalkriterium hat bei der Auswahl der informatischen Grundkonzepte keine große Bedeutung, da das in dieser Arbeit entwickelte Unterrichtskonzept sich lediglich auf die Klassenstufe 5 beschränkt.

Mit der Auswahl der informatischen Grundkonzepte erhebt die Autorin keinen Anspruch auf Vollständigkeit. Auf die explizite Nennung des gesellschaftlichen Bereichs der informatischen Bildung wurde verzichtet, da die Themen entweder in Klassenstufe 5 im Kontext fachlicher oder anderer informatischer Inhalte angesprochen werden oder ihre Relevanz für die Schülerinnen und Schüler in diesem Alter noch sehr gering ist.

Horizontalkriterium

In den Ansätzen der Didaktik der Informatik von Schubert und Schwill (2004) und Nievergelt (1995) wird die *Algorithmisierung* bzw. *Algorithmik* als übergeordnete Kategorie gewählt. Ihr wird dadurch eine besondere Bedeutung zugemessen. Friedrich (2003) nennt sie explizit und ordnet sie der Kompetenzstufe *Verständnis von Konzepten der Informatik* zu. Wedekind & al. (2004a) beschreibt einen *Algorithmus* als Ablauf oder Vorgang eines Schemas. Allein Hubwieser (2001) und Puhlmann (2003) nennen die *Algorithmisierung* nicht. Sie kann höchstens dem Bereich *Verarbeitung von Repräsentation* und der Kompetenzklasse *Gestaltung* zugeordnet werden. Die *Automatisierung* spielt bei Hubwieser die größte Rolle. Explizit wird auf die Möglichkeit der automatischen Verarbeitung von Informationen durch Informatiksysteme hingewiesen. Wedekind bettet das *Automatisieren* in den Kontext der *Schemata* ein. In den anderen Ansätzen kann die *Automatisierung* dem Bereich der *Ablaufprozesse* oder den *prinzipiellen und praktischen Möglichkeiten und Grenzen von Informatiksystemen* zugeordnet werden. Die *Codierung* wird von keinem der Fachdidaktiker erwähnt.

Die *informatische Modellierung* umfasst die Bereiche *Datenstrukturen*, *Modellierung* und *Objektorientierung*. Der Kategorie *Datenstrukturen* werden die Bezeichnungen (abstrakte) *Datentypen* und *Schema* hinzugefügt. Im Komplex des *Programmierens im Kleinen* sind implizit *Datenstrukturen* enthalten, so dass auch hier eine fast vollständige Abdeckung aller fachdidaktischen Positionen gegeben ist. Bei Hubwieser, Friedrich und Puhlmann nimmt das *Modellieren* eine zentrale Rolle ein. Hubwieser bezeichnet die Modellierung als

„[...] einen Themenbereich der Informatik, der aufgrund seiner immensen Bedeutung für die Allgemeinbildung [...] in keinem Informatikunterricht übergangen werden kann.“ (Hubwieser, 2001, S. 85)

Friedrich ordnet das *Anwenden des Modellbegriffs* und das *Entwickeln, Implementieren und Bearbeiten von Modellen* seinen Kompetenzstufen III bis V zu und Puhlmanns Kompetenzklasse *Gestaltung* wird der Modellierungsprozess zugrunde gelegt. Bei Nievergelt kann man die *informatische Modellierung* dem *Entwurf von Informatiksystemen* zuordnen. Ähnlich wie bei der *Codierung* ist auch die *Objektorientierung* in den meisten Ansätzen nicht zu finden, sondern muss als Teil der Themenkomplexe *Programmieren im Großen*, *Strukturen von Informatiksystemen verstehen*, *Modelle entwickeln* oder *Objektorientierte Modellierung* aufgefasst werden. Allein Schwill und Wedekind erwähnen mit der *Vererbung* und den Themenblöcken *Schema und Ausprägung* und *Namensgebung und Kennzeichnung* objektorientierte Konzepte.

Den informatischen Grundkonzepten *Modularisierung*, *Hierarchisierung*, *Abstraktion*, *Iteration* und *Rekursion* wird vor allem bei Schwill große Beachtung geschenkt. Hubwieser verwendet keinen dieser Begriffe. Die *Modularisierung* und *Hierarchisierung* kann bei ihm vielleicht dem Themengebiet *Dekomposition in Subsysteme* und die *Iteration* und *Rekursion* den *zeitlichen Abläufen* zugeordnet werden. Bei Nievergelt können die ersten beiden Bereiche in das Stockwerk *Programmieren im Großen* und die letzten drei in *Programmieren im Kleinen* des *Informatik-Turms* eingeordnet werden. Bei Friedrich trifft die Kompetenz der *begründeten Auswahl von Arbeitsmethoden* diese Kategorie am besten und bei Puhlmann können die fünf Punkte dem *Verständnis der Struktur von Informatiksystemen* zugewiesen werden. Bei Wedekind ist nur das Themengebiet der *Abstraktion* abgedeckt.

Zur Darstellung von Informationen werden die *informatischen Beschreibungsmittel* verwendet. *Syntaxdiagramme* können bei Wedekind in die *Grammatik* und alle anderen in den Bereich der *Schemata* eingeordnet werden. Puhlmann erwähnt überhaupt nichts, Friedrich spricht von der *Darstellung von Handlungen und Abläufen*, bei Nievergelt gibt es im untersten Stockwerk des *Informatik-Turms* Platz und Schwill führt zumindest die *Darstellung der Hierarchisierung* an. Allein bei Hubwieser ist ein ganzer Bereich der *Darstellung von Informationen* gewidmet.

Auch bei den *fachnahen Inhalten* gibt es in der Fachdidaktik Informatik keine einheitliche Position. Schwill und Wedekind stellen die Bedeutung der *Grammatik* mit den Themen *Syntax* und *Semantik* sehr stark in den Vordergrund, Hubwieser spricht die *Syntax* und *Semantik* an und bei Nievergelt gehört das Thema in den Bereich der *Theorie*. Friedrich und Puhlmann hingegen klammern es ganz aus. Dem *Funktionsbegriff* oder dem *funktionalen Programmieren* wird nirgendwo eine Bedeutung beigemessen. Da es sich um eine zur Objektorientierung und dem imperativen Arbeitsstil alternative Denkweise handelt und somit die geistige Flexibilität bei der Lösung von Problemen erhöht, wird sie ebenfalls in den Katalog der informatischen Grundkonzepte aufgenommen.

Zeitkriterium

Betrachtet man die Empfehlungen der BLK, GI, MNU und GDM, die die informatischen Inhalte der Sekundarstufe I festlegen, so ist das Zeitkriterium bei der *Algorithmisierung*, *Automatisierung* und *Modellierung* fast vollständig erfüllt, die *Codierung* und *Objektorientierung* spielt ab ca. 1995 eine Rolle und der Themenkomplex der *Datenstrukturen* ist in den älteren Dokumenten stärker vertreten. Die Inhalte der *Wirkprinzipien von Informatiksystemen* und der *Informatischen Modellierung* sind also längerfristig relevant und haben somit ihre Berechtigung in der Menge der informatischen Grundkonzepte. In allen anderen Bereichen ist dies nicht so offensichtlich.

Modularisierung, *Hierarchisierung*, *Abstraktion*, *Iteration* und *Rekursion* gehören in der GI Empfehlung (2000) in den Bereich *Informatische Modellierung*, im BLK Papier (1995) zum Themenkomplex *Einblick in Wirkungsweise und Produktionsbedingungen von Medien*, in der Beschreibung der Informationstechnischen Bildung der BLK (1987) zu den *Grundstrukturen und Grundbegriffen, die für die Informationstechniken von Bedeutung sind*, und allein in der Empfehlung der MNU (1985a) ist die *Modularisierung* als *Identifikation von Teilproblemen* und *Integration von Teillösungen* sehr deutlich wahrnehmbar.

Der Bereich der *informatischen Beschreibungsmittel* ist wie schon bei der Diskussion des Horizontalkriteriums auf den Darstellungs- und Strukturierungsaspekt von Informationen oder Datenbeständen zurückzuführen, ohne dass in den verschiedenen Empfehlungen konkrete Techniken erwähnt werden. Dort sind Inhalte wie *Darstellung von Algorithmen*, *Verständnis für die Strukturierung von Gegenstandsbereichen*, *Strukturierung komplexer Zusammenhänge*, *Beschreibung und Strukturierung von Modellen* und *Strukturierung umfangreicher Datenbestände* enthalten, so dass das Zeitkriterium also ebenfalls erfüllt ist. Die Themen *Grammatik*, *Syntax*, *Semantik* und der *Funktionsbegriff* sind in den Empfehlungen nicht ausgeführt.

Zielkriterium

Die Schlüsselqualifikationen, die in Abbildung 2.2 über den informationstechnischen Kompetenzen und Inhalten schweben, sind idealisierte Zielvorstellungen einer informatischen Bildung. Sie werden vor allem durch das Erlernen der informatischen Grundkonzepte und nicht durch die *Informationstechnische Grundbildung* gefördert.

Jede kreative Beschäftigung setzt voraus, dass Grundfertigkeiten beherrscht werden. Als informatische Grundfertigkeiten können die *Arbeitstechniken*, die *informatischen Beschreibungsmittel*, ein Vorrat an *Algorithmen* oder auch Kompetenzen im Bereich der *Modellierung* gezählt werden. Die Gestaltungsfähigkeit kann durch das Erlernen von *informatischen Beschreibungsmitteln* ausgebaut, das Abstraktionsvermögen und die Fähigkeit zur Modellierung direkt bei der Anwendung dieser erweitert, die Medienkompetenz durch Grundwissen in den Gebieten der *Automatisierung*, *Modellierung* und verschiedener Darstellungsformen verbessert und das Abschätzen von Technik- und Technologiefolgen durch Kompetenzen im Bereich des Aufbaus von Computersprachen, der *Automa-*

tisierung und *Codierung* unterstützt werden. Vorausschauendes und vernetztes Denken kann durch die Umsetzung von *Algorithmen* in ablauffähige Programme, *Automatisierung* von Abläufen oder *Modellierung* gestärkt werden, vor allem wenn diese Simulationen und somit eine Überprüfung ermöglichen. Organisationsfähigkeit setzt Strukturierung voraus. Werkzeuge dafür enthält der Bereich der *informatischen Beschreibungsmittel*. Kommunikation erfordert eine gemeinsame Sprache für einen bestimmten Problembereich. Sprachmittel können abstrakte Darstellungsformen wie *Syntaxdiagramme*, *Programmablaufpläne*, *UML-Klassendiagramme*, *Hierarchien* und *Netzwerke* sein. Besonders die *fachnahen Inhalte* unterstützen die Lernfähigkeit in anderen Fächern dadurch, dass bestimmte Aspekte eines Lerngegenstands nicht nur abstrakt gelernt werden, sondern als Arbeitsergebnisse sichtbar gemacht werden können. Teamfähigkeit kann durch die informatischen Grundkonzepte nicht unterstützt werden. Trotzdem ist das Zielkriterium für fast alle informatischen Grundkonzepte erfüllt.

Sinnkriterium

Wird ein Informatiksystem von Schülerinnen und Schülern zum Arbeiten oder Spielen genutzt, so muss dabei ein Bezug zu den informatischen Grundkonzepten hergestellt werden können, damit das Sinnkriterium erfüllt ist. In Klaudt und Wursthorn (2005) werden zahlreiche Bezugspunkte aufgezeigt. Die Erstellung eigener einfacher *Algorithmen* kann Einblicke in die tatsächliche Komplexität von großen Softwaresystemen gewähren. Das Anlegen und die Nutzung von Verteilerlisten beim Schreiben von E-Mails an einen immer gleich bleibenden Personenkreis kann eine Zeitersparnis durch die *Automatisierung* der Adressierung verdeutlichen. Das Wissen über die einem Softwaresystem zugrunde liegenden *Datenstrukturen* unterstützt außerdem eine für ein Problem adäquate Auswahl einer Anwendung. Die Operationen zum Beispiel, die auf Tabellen ausgeführt werden können, unterscheiden sich in Textverarbeitungs- und Tabellenkalkulationssystemen maßgeblich.

Sowohl die *Modellierung* als auch die daraus resultierenden Objektstrukturen beeinflussen die Gestaltung der Benutzeroberflächen von Informatiksystemen. Aus diesem Grund erleichtert das oberflächliche Nachvollziehen des Modellierungsvorgangs eines Informatiksystems das Zurechtfinden in den entsprechenden Menüstrukturen. Die vielen Möglichkeiten der Einstellungen in den Benutzerdialogen sind auf das Prinzip der *Abstraktion* zurückzuführen. Sie machen erst die Flexibilität eines Systems aus, die bei der Arbeit ausgenutzt werden kann. Auch *rekursive Strukturen* sind nicht nur unter der Oberfläche eines Informatiksystems versteckt. Zum Beispiel ist die Ordnerstruktur in der Dateiverwaltung eines Betriebssystems rekursiv aufgebaut.

Die *informatischen Beschreibungsmittel* spielen weniger beim Umgang mit einem Informatiksystem, sondern eher in anderen Fächern eine Rolle. Mit *UML-Klassendiagrammen* können Beziehungen zwischen Klassen auf Basis ihrer Eigenschaften dargestellt werden, *hierarchische Abbildungen* und *Netzwerke* veranschaulichen Abhängigkeiten, mit Hilfe von *Programmablaufplänen* können dynamische Prozesse beschrieben werden und syntaktische Strukturen sowohl von „künstlichen“ als auch von natürlichen Sprachen

können durch *Syntaxdiagramme* standardisiert expliziert werden. Für die *fachnahen Inhalte* eröffnen sich ganz neue Perspektiven. Wird zum Beispiel eine eigene kleine Sprache konstruiert und in einem Informatiksystem implementiert, so bleiben *Syntax* und *Semantik* nicht nur theoretische Konstrukte, sondern es kann mit ihnen gespielt werden. Das dabei gewonnene Wissen kann in den Fachunterricht übertragen werden. Gleiches gilt für *Funktionen*. Erst eine Umsetzung auf ein Computersystem ermöglicht einen konstruktiven Umgang mit dem sonst abstrakten Inhalt.

Die Entwicklung des Unterrichtskonzepts für die Klassenstufe 5 zur Vermittlung der informatischen Grundkonzepte als begriffliche und kognitive Grundlage der in den *Bildungsstandards für Informationstechnische Grundbildung* der Realschule in Baden-Württemberg 2004 geforderten informationstechnischen Kompetenzen und Inhalte (vgl. Abbildung 2.5) wird im folgenden Kapitel 3 beschrieben.

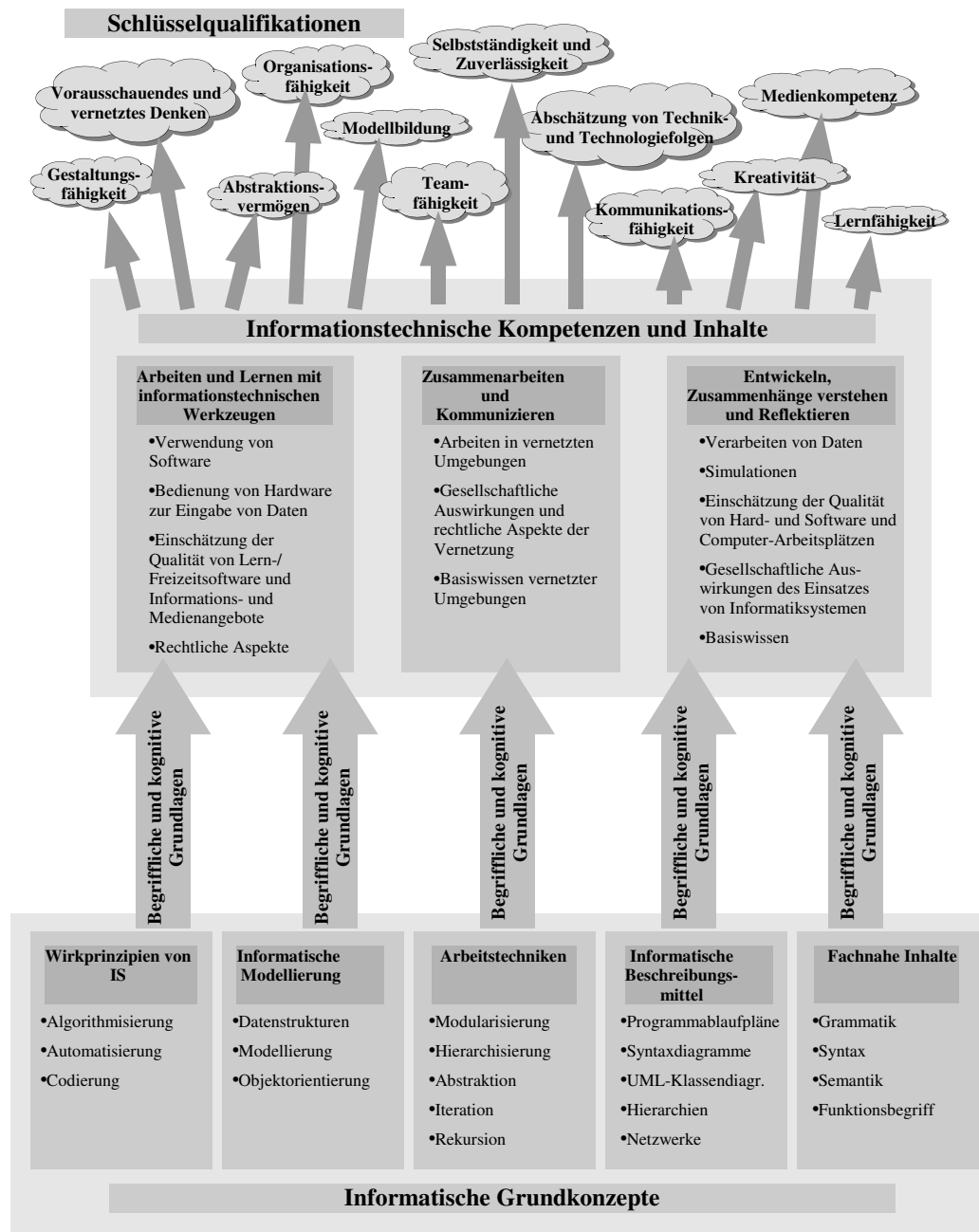


Abbildung 2.5: Informatische Grundkonzepte als begriffliche und kognitive Grundlage informationstechnischer Kompetenzen und Inhalte

3 Konzept des Unterrichtsversuchs

Zu Beginn der Konzeption des Unterrichts stand fest, dass die in Kapitel 2.4 vorgestellten informatischen Grundkonzepte in den Fächern Mathematik, Englisch, Deutsch und Musik in Klassenstufe 5 der Realschule vermittelt und Logo als pädagogisches Werkzeug eingesetzt werden sollte. Logo wurde als Software ausgewählt, da bereits vielfältige und jahrzehntelange positive Erfahrungen bei der Arbeit mit Kindern vorlagen. Ohne lange Einarbeitungsphasen können bereits in Klassenstufe 5 von Beginn an Produkte entwickelt werden. Als universelles Programmierwerkzeug ist es beliebig einsetzbar. Sowohl Teile der Logo-Philosophie (Kapitel 3.1.1) als auch gewisse Spracheigenschaften von Logo (Kapitel 3.1.2), spezielle Metaphern und verschiedene Mikrowelten unterstützen den Lern- und Vermittlungsprozess der informatischen Grundkonzepte. Abbildung 3.1 zeigt die Entwicklung des Unterrichtskonzepts schematisch. Die angegebenen Nummern beziehen sich jeweils auf die Kapitel dieser Arbeit.

Im ersten Schritt stand die Analyse der Lehrpläne der Fächer Mathematik, Englisch, Deutsch und Musik im Mittelpunkt, um Unterrichtsthemen zu identifizieren, in deren fachlichem Kontext die informatischen Grundkonzepte sinnvoll vermittelt werden konnten. Eine Vorauswahl wird in Kapitel 3.3 dargestellt. In der Feinplanung ergaben sich im Laufe des Schuljahrs einige Änderungen. In Kapitel 4 werden die tatsächlich durchgeführten Unterrichtseinheiten, die dort neu erarbeiteten informatischen Grundkonzepte in Verbindung mit den Inhalten der Fächer und die verwendeten Mikrowelten vorgestellt. Parallel zur Analyse des Lehrplans wurde die Idee einer Rittergeschichte für eine mehrwöchige fächerübergreifende Projektarbeit entwickelt. In dieser sollten die Arbeit an Fachinhalten und informatischen Grundkonzepten mit der Erstellung eines multimedialen Produkts in Logo verknüpft werden. Für diejenigen informatischen Grundkonzepte, für die weder Bezugspunkte zu den Fachinhalten noch im Ritterprojekt gefunden werden konnten, wurden am Ende der ersten Phase des Entwurfs des Unterrichtskonzepts fachunabhängige und für die Schülerinnen und Schüler motivierende Themen ausgewählt. Die Ergebnisse werden in Kapitel 4.3 dargestellt.

Die Grobplanung der Unterrichtsthemen in der zweiten Phase gliederte sich in drei Schritte. Zunächst erfolgte eine Sichtung der fachdidaktischen Veröffentlichungen zur Vermittlung informatischer Konzepte und Logo-Unterrichtsvorschläge. Dann wurden die verschiedenen Rollen der informatischen Grundkonzepte und Logo innerhalb des Unterrichtskonzepts identifiziert. Zuletzt wurden die Unterrichtsmethoden und fächerübergreifenden Aspekte des Unterrichtskonzepts (siehe Kapitel 3.4) herausgearbeitet.

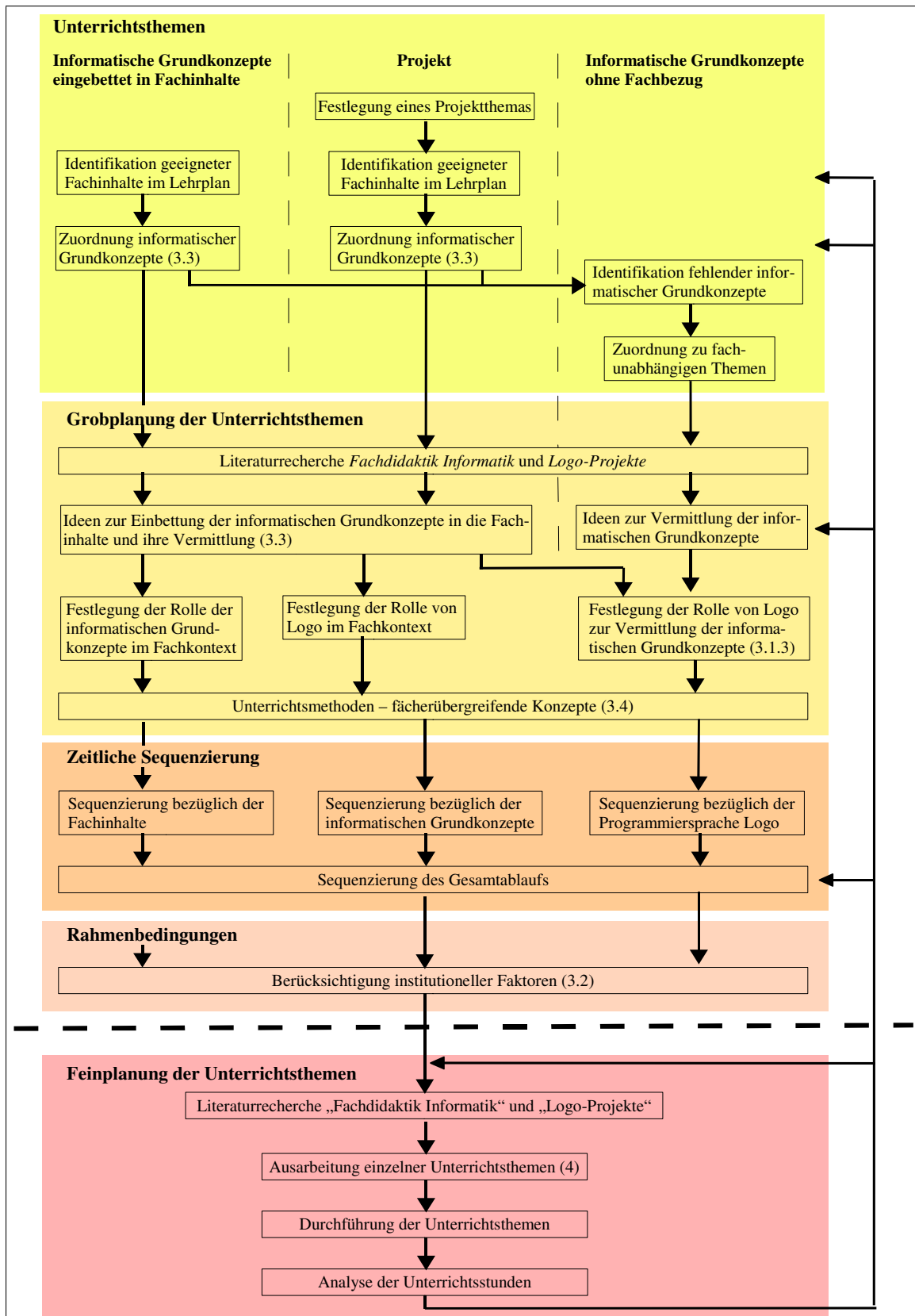


Abbildung 3.1: Entwicklung des Unterrichtskonzepts

Die sich aus der Grobplanung der Themen ergebenden Abhängigkeiten erforderten in der dritten Phase eine Sequenzierung der Themen bezüglich der Fachinhalte, der informatischen Grundkonzepte und der programmiersprachlichen Aspekte von Logo. Aus diesen drei Sequenzen wurde schließlich der Gesamtablauf abgeleitet.

Zwei Monate vor der Erprobung des Unterrichtskonzepts standen die Rahmenbedingungen (siehe Kapitel 3.2) fest. Sie waren in Form von institutionellen Faktoren und durch die eigene Auswahl der beiden Logo-Systeme gegeben. Zum Abschluss der Vorbereitungsphase wurden sie in das Gesamtkonzept integriert. Die konkrete Ausarbeitung der Unterrichtsthemen erfolgte im laufenden Schuljahr (siehe Kapitel 4). Dabei wurden die Erfahrungen aus den einzelnen Unterrichtsstunden in das Gesamtkonzept eingearbeitet, so dass sich im Laufe des Schuljahrs Veränderungen bezüglich der Themen, der Zuordnung der informatischen Grundkonzepte zu diesen, der Methodenwahl zu ihrer Vermittlung und der Reihenfolge der einzelnen Unterrichtseinheiten ergaben.

Parallel zur Entwicklung des Gesamtkonzepts erhielten die Fachlehrerinnen von der Forscherin eine Einführung in die informatischen Grundkonzepte und Logo (siehe Kapitel 3.5). Einige Unterrichtseinheiten konnten mit ihnen in den Veranstaltungen erprobt werden. An vier Abenden zu Beginn des Schuljahrs hatten auch die Eltern die Möglichkeit, sich im Rahmen einer Einführungsveranstaltung in Logo (siehe Kapitel 3.6) über die Inhalte zu informieren.

3.1 Logo

In diesem Kapitel werden die Entwicklung der Programmiersprache Logo, ihre lerntheoretischen Wurzeln, die Kernpunkte ihrer Philosophie und ihre programmiersprachlichen Konzepte, die die Vermittlung der informatischen Grundkonzepte in Klasse 5 besonders unterstützen, vorgestellt.

Logo wurde 1967 von Daniel Bobrow und Wallace Feurzeig, die zu diesem Zeitpunkt beide Mitarbeiter der Firma Bolt, Beranek and Newman (BBN) in Cambridge Massachusetts waren, zusammen mit Seymour Papert am Massachusetts Institute of Technology (MIT) entwickelt. Bobrow implementierte die erste Logo-Version, Feurzeig war hauptsächlich für Forschung und Design zuständig und Papert entwickelte die Logo-Philosophie bezüglich des Unterrichtsstils, der mit der Logo-Arbeit verbunden war (Chakraborty & al., 1999). 1970 zog die gesamte Logo-Gruppe an das MIT um. Dort formierte sie sich neu an dem von Seymour Papert und Marvin Minsky gegründeten Institut für Künstliche Intelligenz. Schon die unterschiedlichen Verantwortlichkeiten der Entwickler von Logo zeigt deutlich, dass es sich bei Logo nicht allein um eine Programmiersprache, sondern ebenfalls um eine Philosophie handelt. In den folgenden beiden Kapiteln werden zunächst die von Papert (1982) ausgeführte Logo-Philosophie vorgestellt und dann die für den Unterrichtsversuch relevanten Konzepte der Programmiersprache erläutert.

3.1.1 Logo-Philosophie

Piaget und Künstliche Intelligenz

Die theoretischen Wurzeln der Logo-Philosophie gehen auf Jean Piaget, bei dem Papert von 1958 bis 1963 arbeitete, und das Wissenschaftsgebiet der Künstlichen Intelligenz zurück. Papert beschreibt *piagetsches Lernen* als natürliches, spontanes Lernen von Menschen, die mit ihrer Umgebung interagieren (Papert, 1982, S. 193). Aus diesem Grund ist es wichtig, dass mit Logo Lernumgebungen entwickelt werden können, die auf möglichst natürliche Weise Interaktionen mit den in ihnen enthaltenen Lerngegenständen unterstützen.

Nach Papert (1982, S. 195) fordert Piaget bei der Untersuchung des Lernens, sich auf die Struktur der Lerninhalte zu konzentrieren und nicht den Lernprozess vom Lerninhalt zu trennen. Papert kritisiert allerdings, dass Piaget seinen theoretischen Schwerpunkt auf das innere Geschehen legt, obwohl er davon ausgeht, dass die inneren Strukturen in Wechselwirkung mit der äußeren Welt stehen. Papert geht einen Schritt weiter. Er sucht in den jeweiligen Bereichen nach geeigneten Lernumgebungen, in denen die natürliche Umgebung keine reichlichen Quellen bietet, damit Kinder ihre intellektuellen Strukturen entwickeln können (Papert, 1982, S. 198). Solche Lernumgebungen bezeichnet er als *Mikrowelten*. Seine bekannteste Mikrowelt ist die des Logo-Igels, in der bereits sehr junge Kinder auf Entdeckungsreise in die Geometrie gehen können.

Neben den Ideen von Piaget beeinflussten auch die Ergebnisse der Forschung im Bereich der *Künstlichen Intelligenz* (KI) als Kognitionswissenschaft die Entwicklung der Logo-Philosophie.

„Das Ziel der KI ist, Ideen über das Denken, die zuvor abstrakt, sogar metaphysisch schienen, konkrete Form zu geben.“ (Papert, 1982, S. 195)

Wenn sich nach Meinung Paperts (1982, S. 43) Kinder mit KI beschäftigen, werden sie dazu angeregt, sich über ihre eigenen geistigen Prozesse Gedanken zu machen, da diese eine konkrete Form erhalten. Sie werden dadurch zu *Erkenntnistheoretikern*. Am intensivsten ist die Auseinandersetzung bei der Artikulation der geistigen Prozesse (Papert, 1982, S. 195).

Die Kommunikation wird in Logo durch die Aufteilung von Problemen in Module erleichtert (Papert, 1982, S. 210). Aus diesem Grund spielen die computertechnischen Strukturen der Prozeduren, die als Module eingesetzt werden können, in der Logo-Philosophie eine besondere Rolle. Neben der Betrachtung einzelner Module ist auch ihre Vernetzung wichtig. Mit Hilfe einer Sprache, die von Kindern leicht erlernt werden kann, muss die Kommunikation zwischen den konstruierten Einheiten beschrieben werden können. Jede Mikrowelt enthält eine solche Sprache, die für den in ihr abgebildeten Lernbereich Vokabeln enthält.

Konstruktionismus

Die Forderungen nach Möglichkeiten der Interaktion oder Reflexion der eigenen Denkprozesse legt nahe, dass die Logo-Philosophie auf einer konstruktivistischen Lerntheorie basiert. Die Kinder programmieren den Computer und dieser programmiert nicht die Kinder (Papert, 1982, S. 28). Zusätzlich ist für Papert wichtig, dass beim Lernen ein Produkt entsteht, das für den Lernenden persönlich bedeutsam wird. Für ihn stellt dies eine Erweiterung des Konstruktivismus dar, die er als *Konstruktionismus* bezeichnet und folgendermaßen charakterisiert:

„Yes doing is a good way to learn. And it is made better by talking and thinking. But we learn best of all by the special kind of doing that consists of constructing something outside of ourselves.“ (Papert, 1999, S. 13)

Wichtig ist, dass ein Arbeitsergebnis in der Realität entsteht, das anderen gezeigt, über das diskutiert und das überprüft werden kann. Dies fördert nach Meinung Paperts das Nachdenken über die Vorstellungen, die zur Entstehung des Produkts beigetragen haben. Diese können dann genauso wie das Arbeitsergebnis selbst angepasst und verbessert werden. Da ein Produkt meist in mehreren Schritten erstellt wird, spielen auch die Möglichkeiten der Interaktion in der Logo-Philosophie eine besondere Rolle. Wichtig ist ein schnelles Feedback des Systems, das durch seine Implementierung als Interpretersprache gesichert ist.

Mikrowelten

Die Konstruktion von Produkten mit Hilfe des Computers ist nicht auf ausgewählte Wissensbereiche beschränkt.

„Der Computer ist der Proteus der Maschinen. Sein Wesen ist seine Universalität, seine Simulationsfähigkeit. Weil er tausend Formen annehmen und tausend Funktionen erfüllen kann, kann er tausend verschiedenen Ansprüchen genügen.“ (Papert, 1982, S. 24)

Diese Universalität zeigt sich in ganz verschiedenen Mikrowelten, die in Logo erstellt werden können. Eine Mikrowelt kann folgendermaßen charakterisiert werden:

„A microworld is a well-defined, but limited learning environment in which interesting things happen and in which there are important ideas to be learned.“ (Goldenberg, 1982, S. 218)

Das Konzept der Beschränkung auf einen sehr kleinen Wissensbereich, in dem nur bestimmte Fragen relevant sind und andere nicht, wurde aus der *Künstlichen Intelligenz* übernommen. Durch die Festlegung von Annahmen und Einschränkungen kann das pädagogische Problem der vernetzten Strukturen von Wissen, das für interessante Inhalte oftmals ein großes Vorwissen erfordert, gelöst werden.

„In diesen Welten können Lernerfahrungen stattfinden, unbehindert von den Komplexitäten der Welt.“ (Papert, 1998, S. 66)

Papert (1982, S. 158 ff.) formuliert am Beispiel der Newtonschen Bewegungsgesetze noch drei weitere Kriterien für das Design von Mikrowelten:

- Eine Mikrowelt muss ein einfaches und sehr zugängliches Beispiel enthalten, mit dem man arbeiten kann,
- es müssen Möglichkeiten für Aktivitäten und zum Spielen angeboten werden und
- alle notwendigen Begriffe müssen innerhalb der Mikrowelt definierbar sein.

Mikrowelten werden dann zu „Brutkästen für Wissen“ (Papert, 1982, S. 153) oder können als „Gewächshaus für eine bestimmte Spezies schlagkräftiger Ideen oder intellektueller Strukturen“ (Papert, 1982, S. 157) genutzt werden.

Die Lernenden gehen in den Mikrowelten auf Entdeckungsreise, erforschen deren Ideen, verallgemeinern diese und entwickeln daraus eigene Theorien. Dies geschieht in einem ganz persönlichen Prozess, der nicht von Dritten übernommen werden kann. Außerdem wird in Mikrowelten nicht allein das Lernen von Inhalten unterstützt, sondern den Lernenden Möglichkeiten angeboten, als Mathematiker, Dichter, Künstler etc. zu agieren. Dies setzt spezielle Sprachen voraus, mit denen wie in einem „Mathematikland“ das „Mathematik Sprechen“ oder in einem „Musikland“ das „Musik Sprechen“ simuliert wird. Die Inhalte der Mikrowelten werden wie eine natürliche Sprache in dem entsprechenden Land erlernt.

Schlagkräftige Ideen

Mikrowelten, die nicht nur konstruktives Arbeiten der Lernenden fördern, sondern gleichzeitig noch den Kontakt zu schlagkräftigen Ideen (engl. powerful ideas) herstellen, sind für das Lernen besonders wertvoll, da sie als Übergangsobjekte für den persönlichen Erwerb der Ideen fungieren (Papert, 1999, S. 13). Eine schlagkräftige Idee definiert Goldberg (1982, S. 224) als

„An idea that can be used in a variety of personally meaningful contexts, that can be thought through, and that is a model for clear thinking is certainly powerful.“

Eine Auflistung von schlagkräftigen Ideen ist bei Papert allerdings nicht zu finden. Er formuliert lediglich die folgenden vier Kriterien:

„They should be simple, general, useful, and syntonetic.“ (Lawler, 1982, S. 150, zitiert)

Einfachheit ist notwendig, da die Ideen verstanden werden müssen. Nützlich sind sie, wenn sie in verschiedenen Kontexten eingesetzt werden können. Dazu müssen sie allgemeingültig sein. Außerdem ist es wichtig, dass schlagkräftige Ideen in Verbindung zu bereits existierendem Wissen der Lernenden stehen, damit persönliche Beziehungen zu

ihnen aufgebaut und Verknüpfungen hergestellt werden können. Watt und Watt (1986) formulieren zwanzig konkrete schlagkräftige Ideen. Sie weisen jedoch auch darauf hin, dass Lernende die Auflistung bei der eigenen Arbeit mit Logo automatisch erweitern.

Zu den schlagkräftigen Ideen, die in dieser Arbeit eine besondere Rolle spielen, gehören das Problemlösen, das strukturierte Programmieren, das Debugging, die Spracherweiterung und die Systematik, die im Folgenden diskutiert werden.

Problemlösen

Problemlösen heißt, einen unerwünschten Anfangszustand in einen erwünschten Zielzustand zu überführen. Zwischen beiden Zuständen liegen Barrieren, die überwunden werden müssen. Wissen und Verfahren müssen auf verschiedenartige Lösungsmöglichkeiten transferiert werden (siehe Gudjons 1999, S. 231). Edelmann (2000, S. 317) und Wessels (1994, S. 356) unterscheiden folgende Problemlösestrategien:

- Versuch und Irrtum
- Problemlösen durch Umstrukturieren
- Problemlösen durch Anwenden von Strategien
- Problemlösen durch Systemdenken
- Problemlösen durch Kreativität.

Lernen in der Schule erfolgt häufig durch Vermittlung von Wissen und Regeln, dem Stellen von Aufgaben und dem schematischen Verwenden des Wissens und der Regeln zur Lösung der Aufgaben. Sollen die Lernenden jedoch als Mathematiker, Dichter oder Musiker arbeiten, so können sie nicht nur Aufgaben oder bereits gelöste Probleme studieren, sondern müssen sich mit neuen Fragestellungen beschäftigen (Goldenberg, 1982, S. 224). Mikrowelten mit Möglichkeiten zur Exploration fördern die Problemlösekompetenz. Angstfrei können die Lernenden verschiedene Lösungswege ausprobieren, diese im Debugging-Modus in nachvollziehbare kleine Einheiten zerlegen, konkrete Objekte manipulieren, unmittelbares Feedback vom System erhalten und daraufhin ihre Vorgehensweisen und ihre kognitiven Strategien anpassen und die Funktionalität der Mikrowelten erweitern. Durch diese Erweiterungen und Umstrukturierungen können sie in kreativen Prozessen neue Lösungsstrategien entwickeln. Wichtig dabei ist, dass sie immer das Gefühl der Kontrolle über das System behalten.

Zusätzlich zum Erwerb von Problemlösekompetenzen werden die Lernenden schon fast gezwungen, eigene Fragestellungen (engl. problem-posing) zu entwickeln, da sie immer wieder auf unvorhergesehene und zunächst unlösbare Probleme stoßen, deren Bewältigung sie in Richtung der Lösung des Gesamtproblems führen. Harper stellt folgende Behauptung auf:

„The development of strategies for problem posing und problem solving is a major goal of Logo.“ (Harper, 1989, S. 11)

Masterson (1985) bezeichnet das Ausmaß, mit dem eine Programmiersprache das Nach-

denken über Probleme erleichtert, als *kognitive Effizienz* (engl. cognitive efficiency). Logo besitzt eine hohe kognitive Effizienz, da in gelungenen Mikrowelten auf einfache Art und Weise Hypothesen bezüglich der Zusammenhänge innerhalb von Problembereichen repräsentiert und ausgetestet werden können.

Ob nun tatsächlich Problemlösekompetenzen durch die Arbeit mit Logo verbessert werden, kann nicht eindeutig belegt werden. Es gibt nach Harper (1989, S. 10) sowohl Studien, die dieser These widersprechen als auch welche, die einen Nachweis erbringen. Den Effekt schätzt Harper (1989, S. 13) mit der folgenden Aussage realistisch ein:

„Logo may not have dramatic effect on the formal thinking abilities of children, but it smooths their acquisition of abstract reasoning.“

Strukturiertes Programmieren

Eine weitere schlagkräftige Idee der Logo-Philosophie ist das strukturierte Programmieren. Der prozedurale Sprachaufbau von Logo legt diese Vorgehensweise beim Problemlösen sehr nahe. Es handelt sich dabei um eine Programmiermethode, bei der komplexe Probleme in eine Vielzahl von Teilproblemen zerlegt werden. Dieser Prozess wird so lange rekursiv fortgesetzt, bis die Lösungen einfach erstellt werden können. Sie werden dann zusammengesetzt, in Prozeduren verpackt und bilden die Grundlage der Gesamtlösung. Diese Teillösungen müssen entsprechend ihrer Funktionalität benannt werden. Papert (1982, S. 134) zitiert einen Jungen, der von „geistgerechten Bissen“ spricht. Er war vor der Anwendung des strukturierten Programmierens mit der Erstellung von Lösungen für komplexere Probleme überfordert gewesen. Papert stellt folgende Behauptung auf:

„Es ist möglich, ein verzweigtes intellektuelles System aufzubauen, ohne jemals einen unverständlichen Schritt zu tun. Und wenn man es hierarchisch strukturiert, kann man das System als Ganzes erfassen, d.h. man kann das System „wie von oben“ sehen.“ (Papert, 1982, S. 135)

Der Vorteil einer strukturierten Problemlösung wird am deutlichsten, wenn bei der Fehlersuche die Orientierung verloren geht. In strukturierten Programmen kann jede einzelne Prozedur auf Korrektheit überprüft werden, bevor die Vernetzung der einzelnen Bausteine zur Gesamtlösung verifiziert wird. Dies erleichtert die Eingrenzung von Fehlerquellen erheblich.

Debugging

Unter Debugging wird klassischerweise die Fehlerverfolgung und -beseitigung verstanden (Claus und Schwill, 2003). Der Begriff *Fehler* muss jedoch sehr weit gefasst werden, da es sich teilweise nicht um Fehler, sondern um unerwartete Verhaltensweisen handelt. Im Gegensatz zu den Erfahrungen der Schülerinnen und Schüler im Schulalltag, in dem Fehler mit schlechten Noten *bestraft* werden, stellen Fehler, auch *Bugs* genannt, in der

Logo-Philosophie eine Quelle zum Lernen dar, da der Prozess der Fehlersuche dazu beiträgt, ein Problem zu verstehen.

„Fehler helfen uns, weil sie uns dazu bringen, herauszufinden, was geschehen ist, zu verstehen, was falsch gelaufen ist und aufgrund dieses Verständnisses die Fehler zu korrigieren.“ (Papert, 1982, S. 146)

Sie stellen eine Möglichkeit dar, die Unstimmigkeiten in den eigenen Vorstellungen aufzudecken, Ideen zu hinterfragen und diese dann durch exakteres Wissen zu ersetzen (Watt und Watt, 1986, S. 1). Debugging in der Logo-Philosophie wird als das Studieren von Fehlern und das Nachdenken über ihre Ursachen definiert. Der in einer Programmiersprache implementierte Debugging-Mechanismus kann den Abbau der Widerstände der Lernenden gegenüber dem Debugging fördern. Wichtig ist, dass der Ablauf eines Programms und die Zustände des Systems zu ausgewählten Zeitpunkten anschaulich visualisiert werden können. Das Debugging beschränkt sich dann nicht nur auf das mentale Nachvollziehen des Programms. Die Effizienz des Lernens ist im Gegensatz zum „Lernen wie Ratten in einem Irrgarten“ (Papert, 1982, S. 145) durch ablaufende Reflexionsprozesse deutlich höher. Goldenberg (1982) fordert, dass die Lehrenden die Lernenden ermutigen sollten, ihre Fehler zu lieben, da sie durch unerwartetes Verhalten neue Ideen entwickeln können, die viel interessanter als die ursprüngliche Aufgabe sind.

„A bug can be neat! It can certainly teach you something ...study it!“
(Goldenberg, 1982, S. 214)

Sicherlich gibt es nicht erst seit der Erfindung des Computers Debugging-Strategien, allerdings können sie mit seiner Hilfe und einem geeigneten System wie Logo nicht nur gedanklich, sondern auch am Bildschirm sichtbar nachvollzogen werden. Diese Debugging-Philosophie wird in Logo-Systemen dadurch unterstützt, dass sich die Interaktion mit dem System an natürlichen Dialogen orientiert und nicht über „Geheimcodes“ erfolgt. Dies gilt auch für Fehlermeldungen. „Deshalb sind Logo-Fehlermeldungen semantisch ausdrucksstark und gezielt [...]“ (Hoppe und Löthe, 1984, S. 7).

Spracherweiterung

Die Kommunikation zwischen Lernenden und dem Computer erfolgt natürlich auch in Logo nicht auf Basis einer zwischenmenschlichen Kommunikation. Lernende müssen die Sprache des Computers erwerben oder diese den eigenen Wünschen anpassen. Durch den bereits weiter oben erwähnten prozeduralen Aufbau von Logo kann das Vokabular sehr leicht erweitert werden. Bei der Definition neuer Wörter als Prozeduren kann auf bereits vorhandene Prozeduren zurückgegriffen werden, da diese als unabhängige Bausteine einsetzbar sind. Durch die Festlegung eigener Namen für Prozeduren, die dann zu neuen Wörtern der gemeinsamen Sprache mit dem Logo-System werden, wird eine persönliche Beziehung zur Arbeit aufgebaut. Der Lernende bringt dem Computer etwas bei, er modelliert die Kommunikation zwischen sich und dem Computer und legt somit das Kommunikationsverhalten fest. Es wird ihm das Gefühl vermittelt, dass er der Be-

stimmende ist, die Kontrolle in der Hand hält und nicht passiv von außen manipuliert wird. Er formt und modifiziert durch neue Wörter die Wirklichkeit, in der er arbeitet.

Systematik

Zuletzt wird noch auf die Bedeutung der schlagkräftigen Idee des systematischen Vorgehens in Logo eingegangen. Ein Computerprogramm läuft mechanisch Schritt für Schritt nach einem programmierten Ablauf ab. Papert (1982, S. 52) behauptet, dass manche Kinder mit dem Erlernen von formalen Lerngegenständen Schwierigkeiten haben, weil sie die Vorteile dieser schematischen Denkweise nicht verstehen. Eine Einsicht können sie nur dadurch gewinnen, dass sie mit Problemstellungen konfrontiert werden, die die Überlegenheit des systematischen Vorgehens anschaulich demonstrieren, und Formalismen in die Hand bekommen, die eine präzise Beschreibung der Systematik ermöglichen. Unabhängig vom Computer oder einer Programmiersprache muss Material entwickelt werden, das zum systematischen Vorgehen anregt, da die Kultur arm an Modellen für dieses Vorgehen ist und Probleme in der Regel eher tastend angegangen werden (Papert, 1982, S. 47). Vor allem die deskriptiven Formalismen und das Prozedurkonzept von Logo tragen dazu bei, systematisches Vorgehen beim Erarbeiten von Lösungen zu erzwingen. Die erarbeiteten Formalismen können getestet werden und sichtbare Ergebnisse entstehen.

Aus den Beschreibungen der verschiedenen schlagkräftigen Ideen wird deutlich, dass sie untereinander stark vernetzt sind. Dies fördert ihr Erlernen. Wichtig ist, dass die Ideen für die Schülerinnen und Schüler bei der Arbeit mit Logo zu Werkzeugen des Denkens werden, so dass sie im Alltag ganz natürlich eingesetzt werden.

3.1.2 Logo als Programmiersprache

Logo, auf seine programmiersprachlichen Aspekte reduziert, ist ein Dialekt der Programmiersprache LISP (List Processing Language), die vor allem im Bereich der *Künstlichen Intelligenz* in der Forschung eingesetzt wird. Der Einstieg in die Arbeit mit Logo ist jedoch sehr viel leichter, da sich die Syntax stärker an der Umgangssprache orientiert (Ziegenbalg, 1985, S. 5).

Harvey (1982, S. 191) beschreibt das Designziel von Logo folgendermaßen:

„Logo is designed to make explicit many of the fundamental ideas of computer programming. Someone who learns Logo is likely to have a very clear idea of the nature of variables, procedures, and most other programming constructs.“

Logo-Systeme enthalten ca. 200 bis 400 Prozeduren aus verschiedenen Bereichen wie zum Beispiel den Grundrechenarten (SUMME, DIFFERENZ, ...), der Logik (UND, ODER, ...) oder der Ein- und Ausgabe von Daten (LIESWORT, DRUCKZEILE, ...). Diese werden auch als

Logo-Grundprozeduren (engl. primitives) bezeichnet. Die bekanntesten Befehle sind die der Igelgrafik (engl. turtle graphic) wie zum Beispiel **VORWÄRTS**, **RÜCKWÄRTS**, **RECHTS**, und **LINKS**, mit denen ein Igel auf dem Bildschirm herumkommandiert werden kann. Sie werden häufig als Einstieg in das Programmieren mit Logo verwendet. Abbildung 3.2 zeigt den Igel als Dreieck, der einen Stift in der Hand hält und bereits 100 Schritte in Richtung seiner Nase nach vorne gegangen ist, sich dann um 90 Igeleinheiten nach rechts gedreht hat und schließlich noch seinen Weg um 50 Schritte fortgesetzt hat.

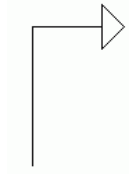


Abbildung 3.2: Zeichnung des Logo-Igels

In Logo wurden ihm dafür folgende Befehle erteilt:

```
VORWÄRTS 100  
RECHTS 90  
VORWÄRTS 50
```

Befehle sind Prozeduren, die eine Nebenwirkung auf das System haben und dessen Zustand verändern. Mit Hilfe dieser kann ein Programmierer das System, im Fall der Igelgrafik den Igel, zu einer Aktion veranlassen. Zur Erweiterung können eigene Befehle erstellt werden. Sie fassen eine Folge von Prozeduren unter einem neuen Namen zusammen und können analog zu den Grundbefehlen verwendet werden. Als Bezeichnung wird in der Regel ein Imperativ gewählt, der die Funktionalität inhaltlich möglichst präzise beschreibt. Ein Beispiel dafür ist der Befehl **Quadrat**, durch dessen Aufruf der Igel am Bildschirm ein Quadrat zeichnet. Abgekürzt wird er nicht mit der Imperativform „zeichne ein Quadrat“, sondern mit **Quadrat**.

```
PR Quadrat  
  WIEDERHOLE 4 [ VORWÄRTS 40  RECHTS 90 ]  
ENDE
```

Wird der Befehl **Quadrat** aufgerufen, so wiederholt der Igel viermal das Vorwärtsgehen von 40 Schritten, gefolgt von dem Rechtsdrehen um 90 Igeleinheiten. Am Bildschirm entsteht das Quadrat in Abbildung 3.3.



Abbildung 3.3: Quadrat des Logo-Igels

Um den Befehl flexibler zu gestalten, kann ein Parameter *Länge* eingeführt werden.

```
PR Quadrat :Länge
  WIEDERHOLE 4 [ VORWÄRTS :Länge RECHTS 90 ]
ENDE
```

Der Igel wiederholt nun viermal das Vorwärtsgen um eine bestimmte *Länge* und das Rechtsdrehen um 90 Igeleinheiten. Die Längenangabe wird erst beim Aufruf an den Formalparameter übergeben, so dass der Igel weiß, wie weit er gehen muss.

Abstrakt kann ein Befehl als eine Box dargestellt werden, in die oben Eingaben hineinfallen. Abbildung 3.4 zeigt links die Box des Befehls **Quadrat**, die genau einen Eingabetrichter besitzt und eine Eingabe erhält. Die Boxen mit den Trichtern werden auch als *Maschinen* bezeichnet, da sie die Vorgänge, die in ihrem Inneren ablaufen, verbergen.

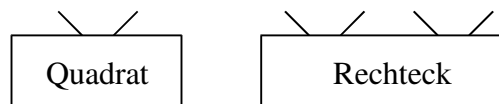


Abbildung 3.4: Maschinendarstellung der Befehle **Quadrat** und **Rechteck**

Die Maschine des Befehls **Rechteck** hat zwei Eingabetrichter, da ein Rechteck zwei unterschiedlich lange Seiten besitzt. Ihr Logo-Programmcode sieht folgendermaßen aus:

```
PR Rechteck :Länge :Breite
  WIEDERHOLE 2 [ VORWÄRTS :Länge RECHTS 90
                VORWÄRTS :Breite RECHTS 90 ]
ENDE
```

Parameter und auch Variablen werden in Logo nicht typisiert. Es können ihnen während eines Programmablaufs also beliebige Werte zugewiesen werden. Dadurch wird der Programmierer nicht gezwungen, eine Variable nur für einen ganz bestimmten Zweck zu deklarieren. Harvey (1982, S. 174) sieht dies nicht als Nachteil an, da eine Variable in Logo in der Regel zu einer Prozedur gehört und somit an deren Kontext gebunden ist.

Im Gegensatz zu Befehlen werden in Logo Prozeduren, die einen Wert an das System zurückgeben, als *Funktionen* bezeichnet. Die Logo-Grundprozedur **SIN** ist zum Beispiel eine Funktion, die für einen im Gradmaß angegebenen Winkel den Sinuswert ermittelt.

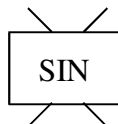


Abbildung 3.5: Maschinendarstellung der Funktion **SIN** (Sinus)

Ihre Maschinendarstellung hat zusätzlich zum Eingabetrichter für den Winkel einen Trichter an der unteren Seite der Box, aus dem symbolisch der Rückgabewert, der entsprechende Sinuswert, der Funktion heraus fällt. Eine Logo-Prozedur kann maximal einen Ausgabetrichter besitzen. Da Funktionen Ein- und Ausgabewerte besitzen, können sie zu Funktionsgeflechten verkettet werden. Ein zufälliger Satz aus einem Subjekt und einem Prädikat kann durch die in Abbildung 3.6 dargestellte Komposition der Funktionen **Subjekt**, **Prädikat** und **SATZ** und des Befehls **DRUCKEZEILE** am Bildschirm ausgegeben werden.

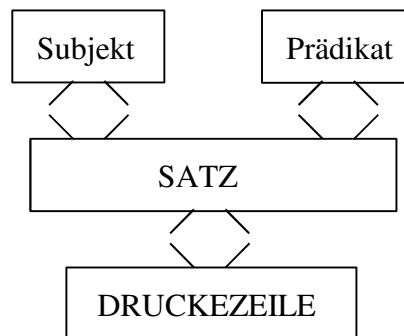


Abbildung 3.6: Funktionsgeflecht zur Ausgabe eines zufälligen Satzes am Bildschirm

Die Funktionen **Subjekt** und **Prädikat** geben zufällig ein Subjekt und Prädikat zurück, mit Hilfe der Funktion **SATZ** können Wörter zu einem Satz bzw. einer Liste zusammengesetzt werden und der Befehl **DRUCKEZEILE** gibt die übergebene Liste am Bildschirm aus. Es ist leicht zu erkennen, wie die Daten zwischen den Funktionen fließen und die Maschinen kommunizieren. Zur Bezeichnung von Funktionen werden häufig Substantive verwendet, die die zurückgegebenen Werte charakterisieren (vgl. **Subjekt**, **Prädikat**). Wird das Konzept der funktionalen Programmierung streng verfolgt, so sind in einem Programm keinerlei globale Variablen erforderlich, sondern die Daten werden zwischen den Funktionen an den Schnittstellen ausgetauscht.

Logo ist eine erweiterbare Sprache. Es werden keine Programme im klassischen Sinne geschrieben, sondern das Logo-Grundsystem wird um Befehle und Funktionen erweitert. Diese müssen, anders als Bibliotheken, nicht extra in Programme eingebunden werden, sondern stehen ohne zusätzlichen Aufwand zur Verfügung und können mit bereits bestehenden Modulen interagieren (Ziegenbalg, 1985, S. 152). Dies gilt genauso für Kontrollstrukturen. Die Fakultät der natürlichen Zahl $N = 8$ könnte in Logo folgendermaßen berechnet werden:

```

SETZE "Fak 1
SETZE "N 8
Solange [:N > 1] [SETZE "Fak  PRODUKT :Fak :N
                  SETZE "N  DIFFERENZ :N 1]
DRUCKEZEILE :Fak
  
```

Nicht alle Logo-Systeme enthalten die Kontrollstruktur **Solange**. Fehlt sie, so muss ein Befehl implementiert werden (Ziegenbalg, 1985, S. 153). Der Befehlsname ist **Solange**, der erste Parameter die Abbruchbedingung der Schleife ($:N > 1$) und der zweite die Handlung, die während eines Schleifendurchlaufs ausgeführt wird (**SETZE "Fak PRODUKT :Fak :N SETZE "N DIFFERENZ :N 1**). Dies ist in Logo nur dadurch möglich, dass Parametern logische Ausdrücke und Befehlsfolgen als Werte zugewiesen werden können. Der Befehl **Solange** hat, wie bereits beschrieben, die beiden Eingabeparameter *Bedingung* und *Handlung*. Zunächst erfolgt die Prüfung (**PRUEFE**) der ausgewerteten (**TUE**) Bedingung. Wenn das Ergebnis *wahr* ist (**WENNWAHR**), wird die Handlung ausgeführt (**TUE**) und der **Solange**-Befehl (**Solange**) erneut aufgerufen.

```
PR Solange :Bedingung :Handlung
  PRUEFE TUE :Bedingung
  WENNWAHR [ TUE :Handlung
              Solange :Bedingung :Handlung ]
ENDE
```

Der Befehl **Solange** ruft sich also selbst auf und hat somit eine rekursive Struktur. Harvey (1982, S. 166) beschreibt den Vorteil von rekursiven Problemlösungen folgendermaßen:

„By allowing a complicated problem to be described in terms of simpler versions of itself, recursion allows very large problems to be stated in a very compact form.“

Dies wird bereits an diesem einfachen Beispiel deutlich. Nicht nur Kontrollstrukturen, sondern auch Datenstrukturen wie z.B. Arrays oder Matrizen können als eigenständige Prozeduren implementiert werden (vgl. Ziegenbalg, 1985, S. 163). Als Datentypen stehen in Logo nur Zahlen, Wörter und Listen zur Verfügung. Zu den Zahlen gehören ganze Zahlen und Dezimalzahlen. Wörter sind Zeichenketten aus Buchstaben, Ziffern und Sonderzeichen. Sie können sowohl als Wörter in ihrem herkömmlichen Sinn als Elemente einer natürlichen Sprache als auch als Namen und somit als Symbole für Variablen und Prozeduren verwendet werden. Die mächtigste Datenstruktur in Logo ist die Liste. Sie kann aus einer beliebigen Anzahl von Zahlen, Wörtern oder selbst wieder Listen bestehen. Ihre Struktur ist beliebig und kann sich während der Bearbeitung verändern. Im Gegensatz zu Arrays können Listen sehr flexibel eingesetzt werden. Der direkte Zugriff auf einzelne Elemente von Listen ist zwar nicht möglich, Zugriffsfunktionen können jedoch implementiert werden. Der Grad der Flexibilität bei der Programmierung mit Logo wird dadurch noch erhöht, dass Prozeduren während des Programmablaufs dynamisch verändert werden können. Dies ist möglich, da Logo intern nicht zwischen Daten und Programmen unterscheidet. Insgesamt ist Logo also im Sinne von Harvey eine sehr mächtige Sprache für Lernende.

„The power of a language is a way of measuring how much the language helps you concentrate on the actual problem you wanted to solve in the first place, rather than having to worry about the constraints of the language.“
(Harvey, 1997, S. XV)

Ein Teil der Mächtigkeit macht die Implementierung als Interpretersprache aus. Da kein Maschinencode generiert wird, entfällt bei der Programmentwicklung die zeitaufwändige Arbeit der Programmeingabe in einen Editor, des Kompilierens und des Ausführens von generiertem Maschinencode. In Logo kann im Direktausführungsmodus ein Befehl eingegeben werden, der sofort bearbeitet wird. Vor seiner Ausführung muss er vom Interpreter übersetzt werden. Somit laufen zwar bereits geschriebene Programme langsamer, die Programmentwicklung und auch das Debugging sind jedoch wesentlich einfacher und schneller. Letzteres spielt beim Erlernen einer Programmiersprache die größere Rolle, da die Lernenden normalerweise die Programme nur zu Übungszwecken entwerfen und später nie wieder verwenden. Ihre Laufzeit stellt also keine relevante Größe dar (Harvey, 1982, S. 166).

Analog dazu verhält es sich mit der Speicherverwaltung. Programmiersprachen, die nicht typisiert sind oder dynamische Datentypen anbieten, haben eine sehr komplexe Speicherverwaltung. Da diese automatisch vom Interpreter übernommen wird, hat ein Programmierer durch diese Flexibilität keine Nachteile. Allein der Zugriff auf die Daten ist etwas langsamer (Harvey, 1982, S. 174).

Zuletzt ist noch wichtig, dass die Syntax von Logo streng zwischen den Namen und den Werten von Variablen unterscheidet. In Logo bedeutet

```
SETZE "X :X + 1
```

„setze unter dem Namen X den Wert von $X + 1$ “. Der Variablenname muss mit einem führenden Anführungszeichen gekennzeichnet werden und auf den Wert einer Variablen wird mit Hilfe des Doppelpunkts zugegriffen. Die Zuweisung erfolgt explizit durch den Befehl **SETZE** (Harvey, 1982, S. 180). In manchen Logo-Versionen gibt es auch die Möglichkeit, Zuweisungen als

```
NAME :X + 1 "X
```

auszudrücken. Der Code kann dann als „nenne das Ergebnis der Addition X“ gelesen werden.

Zusammenfassung

Logo als Programmiersprache bietet beim Problemlösen einige Vorteile. Es minimiert die Zeit, die zum Schreiben der Lösungen notwendig ist, unterstützt die Natürlichkeit der Formulierungen und spiegelt in den Problemlösungen das ursprüngliche Problem wider. Für Lernende sind dies wichtigere Eigenschaften als die Speicher- und Laufzeiteffizienz. Ziegenbalg (1985, S. 179) bezeichnet die Vorteile als *kognitive Effizienz einer Program-*

miersprache. Er legt damit im Gegensatz zu Masterson (1985), bei dem das Nachdenken über Probleme im Vordergrund steht (vgl. Kapitel 3.1.1), mehr Wert auf die Spracheigenschaften. Abelson (1982, S. 112) bekräftigt diese Aussage mit seiner Forderung:

„If we can dispel the delusion that learning about computers should be an activity of fiddling with array indexes and worrying about whether x is an integer or a real number, we can begin to focus on programming as a source of *ideas*. For programming is an activity of *describing* things. The descriptions are phrased so that they can be interpreted by a computer, but that is not really so important. Computational descriptions, like those of science or mathematics, provide a perspective, a collection of „tools of thought“, such as procedural organization, hierarchical structure, and recursive formulations.“

3.1.3 Logo und informatische Grundkonzepte

Die Logo-Philosophie und die programmiersprachlichen Aspekte von Logo unterstützen, wie bereits in den Kapiteln 3.1.1 und 3.1.2 angedeutet, das Erlernen informatischer Grundkonzepte. Eine ausführliche Darstellung erfolgt in diesem Kapitel.

Wirkprinzipien von Informatiksystemen

Jeder Programmcode einer Programmiersprache ist ein *Algorithmus*, eine eindeutige Verarbeitungsvorschrift. Schon in der ersten Stunde mit Logo, in der der Igel ein Quadrat zeichnet (vgl. Kapitel 3.1.2), kann eine Abfolge von Verarbeitungsschritten in der Sprache des Igels formuliert werden. Erfolgt die Eingabe der Befehle im Direktausführungsmodus über die Kommandozeile, so werden sie sofort interpretiert. Das Verhalten des Igels kann also Schritt für Schritt am Bildschirm nachvollzogen werden. Es wird deutlich, dass der Computer keine willkürlichen Entscheidungen trifft, sondern sich genau an die vorgegebenen Anweisungen hält. Gerade zu Beginn der Arbeit ist dieses direkte Feedback für das Verständnis der Arbeitsweise von Informatiksystemen sehr hilfreich. Unterstützt wird es außerdem von dem anthropomorphen Modell des Igels. Seine Bewegungen können die Kinder mit ihrem eigenen Körper imitieren. Werden Algorithmen komplexer, so können sie mit Hilfe von Debugging-Techniken nachvollzogen werden. Logo bietet dafür die in Kapitel 3.1.1 beschriebenen Debugging-Mechanismen an.

Schon zu Beginn der Arbeit mit Logo können Prozesse *automatisiert* werden. Beim Zeichnen eines Dreiecks können die Befehle einzeln, als Folge oder mit dem **WIEDERHOLE**-Befehl im Direktausführungsmodus eingegeben werden. Außerdem kann ein neuer Befehl erstellt werden, der dann direkt ausgeführt wird. Durch die Reduzierung des Programmcodes wird der Automatisierungsgrad erhöht.

Informatische Modellierung

Variablen, Prozedurparameter und Funktionswerte sind in Logo nicht typisiert (vgl.

Kapitel 3.1.2). Aus diesem Grund entstehen bei der Arbeit insgesamt wenige syntaktische Fehler, deren Fehlermeldungen oftmals schwierig zu verstehen sind. Stattdessen wird ein Nachdenken über die inhaltlichen Ergebnisse erzwungen. Soll zum Beispiel der Titel der Personen *Präsident Bush*, *Jimmy* oder *Königin Elisabeth* ausgewählt werden, so können erstaunliche Ergebnisse entstehen. Aus der Liste

```
[[Präsident Bush] Jimmy [Königin Elisabeth]]
```

wird das erste Element der Listeneinträge mit folgender Anweisung am Bildschirm ausgegeben:

```
DRUCKEZEILE ERSTES ZEL [[Präsident Bush] Jimmy [Königin Elisabeth]]
```

Das Ergebnis ist *Präsident*, *J* oder *Königin*. Die Funktion `ZEL` (Zufallselement aus Liste) kann Daten vom Typ `Liste` (`[[Präsident Bush]`, `[Königin Elisabeth]`) und vom Typ `Wort` (`Jimmy`) zurückgeben. Die Funktion `ERSTES` extrahiert aus einer Liste deren erstes Element (`Präsident`, `Königin`) und aus einem Wort dessen erstes Zeichen (`J`). Die Befehlszeile erzeugt keine Fehlermeldung, liefert jedoch im Fall von *Jimmy* nicht das gewünschte Ergebnis. Aufgrund der Diskrepanz zwischen dem erwarteten und dem tatsächlichen Ergebnis werden Denkprozesse angestoßen, um eine Lösung für Personen ohne Titel zu erarbeiten. Eine denkbare Lösung ist

```
DRUCKEZEILE ERSTES ZEL [[Präsident Bush] [Jimmy] [Königin Elisabeth]]
```

Andere Lösungen entstehen beim weiteren Experimentieren. Da für alle Datentypen in den Igel-, Wort- und Satz-Mikrowelten Anwendungsmöglichkeiten zur Verfügung stehen, die direktes Feedback zurückgeben, werden Schülerinnen und Schüler durch die geschilderte Problematik nicht überfordert. Gleichzeitig können mit Hilfe des äußerst flexiblen Datentyps der Liste auch sehr komplexe und der Problemstellung angepasste eigene *Datenstrukturen* entwickelt werden.

In jedem Logo-System kann der Igel als ein Objekt einer Igel-Klasse betrachtet werden. Er besitzt einen Zustand, der durch seine Position auf dem Bildschirm, seine Blickrichtung, die Haltung seines Stiftes etc. beschrieben wird. Außerdem gibt es verschiedene Befehle und Funktionen wie `VORWÄRTS`, `LINKS`, `STIFTAB`, `STIFTHOCH`, `AUFKURS`, `AUFXY`, `KURS`, `ORT` oder `STIFTUNTEN`, mit denen der Zustand des Igels gelesen oder verändert werden kann. Das Konzept der Attribute und Methoden kann leicht nachvollzogen werden. In objektorientierten Logo-Systemen spielt des Weiteren die Namensgebung eine wichtige Rolle. Jedes Objekt muss mit seinem Namen angesprochen werden. Objekte verschiedener Klassen besitzen außerdem ganz unterschiedliche Eigenschaften und Methoden. Ein Schaltflächen-Objekt hat wie ein Igel-Objekt Attribute, die seine Position auf dem Bildschirm beschreiben. Zusätzlich besitzt es Eigenschaften, die den Text auf der Schaltfläche oder seine Breite und Höhe festlegen. Entsprechend gibt es Methoden wie `SETCAPTION` oder `SETSIZE`, mit denen die Attributwerte gesetzt werden können. Vorteilhaft wirkt sich in Logo aus, dass viele Objekte an der Oberfläche sichtbar sind, ein

großer Teil ihrer Attribute dem intuitiven Verständnis der Objekte aus der realen Welt entsprechen, die Nomenklatur der Methoden einem nachvollziehbaren Schema folgt, die Methoden teilweise über das Kontextmenü ohne Programmierung ausgeführt werden können und nur eine begrenzte Anzahl von Klassen mit dem System ausgeliefert wird. Insgesamt erleichtert dieses System den schwierigen Einstieg in die *objektorientierten Denkweisen*, indem die fundamentalen Konzepte *Klasse*, *Objekt*, *Attribut* und *Attributwert* schrittweise auf unterschiedlich abstrakten Niveaus eingeführt werden können.

Arbeitstechniken

Der streng prozedurale Aufbau der Programmiersprache Logo erfordert eigentlich einen *modularisierten Entwurf* einer Problemlösung. Teillösungen können in einzelnen Prozeduren verpackt und *hierarchisch* zur Gesamtlösung zusammengesetzt werden. Prozeduren ohne globale Parameter sind sogar unabhängig von dem ursprünglichen Problemkontext flexibel einsetzbar und können als Spracherweiterung aufgefasst werden.

Das Prinzip der *Iteration* wird bei der Verwendung des WIEDERHOLE-Befehls deutlich und *rekursive Lösungen* werden durch rekursive Aufrufe von Funktionen und Befehlen ermöglicht. Das Konzept der *Abstraktion* in Logo wurde bereits anhand der Erweiterung des Quadrat-Befehls in Kapitel 3.1.2 vorgestellt.

Informatische Beschreibungsmittel

Programmablaufpläne, Syntaxdiagramme, UML-Klassendiagramme, Hierarchien und Netzwerke sind standardisierte informatische Beschreibungsmittel. In der professionellen Softwareentwicklung werden sie teilweise als Basis zur automatischen Generierung von Programmcode verwendet. Da Logo für diese Art der Codegenerierung keine Unterstützung anbietet, wird ihr Erwerb durch die Programmiersprache selbst nicht gefördert. Allerdings gibt es Bezüge zwischen den grafischen Darstellungen und ihrer Umsetzung in ausführbaren Logo-Programmcode. *Programmablaufpläne* bestehen zum Beispiel nach Claus und Schwill (2003) aus Aktionen und Verzweigungen. In Logo entspricht dies Prozeduraufrufen und dem WENN-Befehl.

Syntaxdiagramme können mit Hilfe der Funktion SATZ umgesetzt werden. Terminalsymbole werden als Zeichen oder Wort und Nichtterminalsymbole als Funktionen, die einen entsprechenden Wert zurückgeben, implementiert. Ein deutscher Satz aus einem Subjekt und einem Prädikat kann durch das Syntaxdiagramm

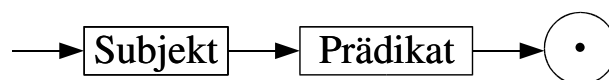


Abbildung 3.7: Syntaxdiagramm eines Satzes bestehend aus Subjekt und Prädikat

repräsentiert und als

DRUCKEZEILE (SATZ Subjekt Prädikat ".")

in Logo umgesetzt werden.

Analog würde das Vorgehen für *UML-Klassendiagramme* bedeuten, dass entsprechende Klassen in Logo entwickelt werden. Dies ist jedoch erst bei größeren und komplexeren Problemstellungen sinnvoll. In Klasse 5 werden UML-Klassendiagramme nur zur Veranschaulichung von Sachverhalten verwendet. Gleiches gilt für *Hierarchien* und *Netzwerke*. Ihre Umsetzung in die Logo-Datenstruktur Liste und die Programmierung von entsprechenden Zugriffsfunktionen auf die Knoten der Hierarchien und Netzwerke überfordern Zehnjährige.

Fachnahe Inhalte

Das Konzept der *Funktionen* kann spielerisch beim Experimentieren mit Wörtern und Sätzen erprobt werden. Die Wort- und Satz-Mikrowelten stellen analog zur Igel-Mikrowelt ein Repertoire an Grundfunktionen für die Definition von *Grammatiken* und die automatische Erzeugung von Sätzen zur Verfügung. Beispielsweise gehören SATZ, WORT, ERSTES, OHNEERSTES, LETZTES, OHNELETZTES, WORT?, SATZ?, DRUCKEZEILE oder ZEL zu diesen Funktionen. Zu Beginn der Arbeit tritt häufig die Fehlermeldung „Was soll denn geschehen mit ...“ auf. Es wird vergessen, dass Funktionen Objekte zurückgeben, die weiterverarbeitet werden müssen. Es wird also immer wieder auf den Rückgabewert aufmerksam gemacht.

Die *Syntax* von Sätzen einer natürlichen Sprache kann in Logo sehr einfach definiert werden. Dies wurde bereits weiter oben bei der Umsetzung der Syntaxdiagramme in Logo-Programme beschrieben. Bei der automatischen Generierung der Sätze wird dabei sehr schnell deutlich, dass zu einer Sprachdefinition nicht nur syntaktische, sondern auch *semantische Regeln* gehören, da teilweise „Unsinnssätze“ entstehen. Logo als Arbeitsmittel verdeutlicht die Verletzung der semantischen Regeln.

3.2 Rahmenbedingungen

Der Unterrichtsversuch zur Vermittlung der informatischen Grundkonzepte wurde über das gesamte Schuljahr 2003/2004 in einer 5. Klasse an einer Realschule in Baden-Württemberg durchgeführt. Die Versuchsklasse wurde vom Rektor zufällig ausgewählt und setzte sich aus 13 Jungen und 17 Mädchen zusammen. Sie besuchten zuvor Grundschulen in sechs verschiedenen Gemeinden, so dass sie sich zu Beginn des Schuljahrs größtenteils nicht kannten.

Unterrichtsfächer

Der Unterrichtsversuch fand in den Fächern Mathematik, Englisch, Deutsch und Musik statt, die von drei Fachlehrerinnen unterrichtet wurden. Eine Lehrerin war für Deutsch

und Musik verantwortlich. In jedem Fach wurden die informatischen Grundkonzepte in einer Stunde pro Woche vermittelt, so dass in Mathematik noch drei, in Englisch und Deutsch vier und in Musik eine weitere Unterrichtsstunde zur Verfügung standen. Letztere wurde für alle Schülerinnen und Schüler der Klassenstufe 5 als Projektstunde im Chor, im Orff-Orchester, in der Band, in der Akrobatik-Gruppe etc. abgehalten, so dass die Versuchsklasse keinen regulären Musikunterricht hatte. Aus diesem Grund mussten alle Fachinhalte aus Musik im Unterricht zu den informatischen Grundkonzepten vermittelt werden.

Unterrichtsorganisation

Im ersten Schulhalbjahr fand der Unterricht jeweils montags in der zweiten und dritten und donnerstags in der ersten und zweiten Stunde statt. Im zweiten Schulhalbjahr gab es an beiden Tagen eine Verschiebung um eine Stunde nach hinten. Während dieser vier Schulstunden unterrichtete die Forscherin die gesamte Klasse. Die entsprechende Fachlehrerin war im Unterrichtsraum anwesend. Zusätzlich arbeitete die Forscherin vom 8. bis 12. Dezember, als eine schulweite Methodikwoche stattfand, mit jeweils der Hälfte der Klasse an drei Tagen zwei Stunden allein im Computerraum. Jeden Donnerstag wurde in der siebten und achten Schulstunde eine freiwillige Computer AG angeboten, in der die Schülerinnen und Schüler sehr frei an eigenen oder vorgeschlagenen Themen arbeiteten.

Unterrichtsräume

In den vier regulären Unterrichtsstunden waren sowohl das Klassenzimmer als auch der Computerraum reserviert. Meistens begann der Unterricht im Klassenzimmer, das sich im dritten Stockwerk befand. Es war weder mit einem stationären Rechner noch mit einem Beamer ausgestattet. Häufig fand in der Doppelstunde ein Wechsel in den Computerraum im zweiten Stockwerk statt. Die Tischanordnung in beiden Räumen ist in Abbildung 3.8 und 3.9 dargestellt.

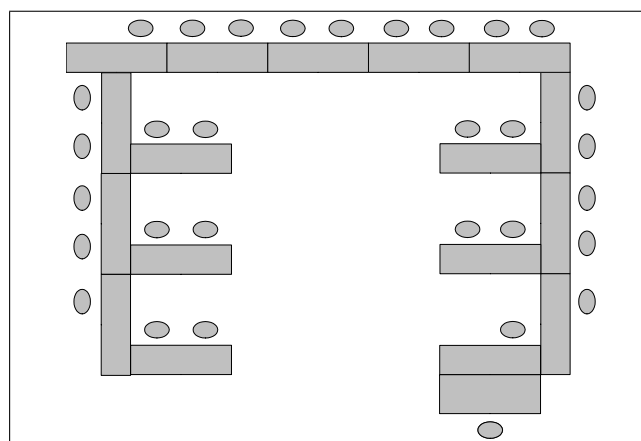


Abbildung 3.8: Klassenzimmer

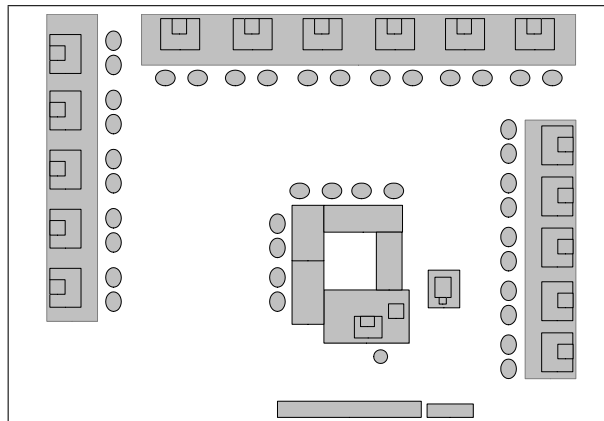


Abbildung 3.9: Computerraum

Im Computerraum befanden sich 16 vernetzte Schüler- und ein Lehrer-Computer. Am Rechner des Lehrers war ein Beamer angeschlossen. Die Arbeitsergebnisse der Schülerinnen und Schüler wurden auf einem zentralen Server und nicht auf den lokalen Rechnern abgespeichert, so dass sie vom Lehrer-Computer aus vor der Klasse präsentiert werden konnten. An zwei Rechnern durften Kinder allein, ansonsten nur in Paaren arbeiten. Über das Schuljahr hinweg veränderte sich die Zusammensetzung der Partner. Kinder, die im Klassenzimmer nebeneinander saßen, bildeten im Computerraum nicht zwangsläufig ein Paar.

Ab Mai standen dem Unterrichtsversuch 15 Convertibles zur Verfügung. Die Funktionalität, das Display auf die Tastatur zu legen und Benutzereingaben handschriftlich mit einem speziellen Stift vorzunehmen, wurde nur in einer Stunde bei der Verwendung eines selbst erstellten Malprogramms genutzt. Aufgrund der Mobilität der Geräte fand der Unterricht ab diesem Zeitpunkt nur noch im Klassenzimmer statt. Die Geräte mussten vor ihrer Verwendung aus dem darunter liegenden Stockwerk geholt werden, da sie im Klassenzimmer nicht sicher eingeschlossen werden konnten. Sie standen also nicht unmittelbar am Arbeitsort zur Verfügung. Aufgrund eines fehlenden Beamers im Klassenzimmer konnten auch keine elektronischen Materialien mehr präsentiert werden.

Logo-Systeme

Während des gesamten Unterrichtsversuchs wurde nur Logo-Software eingesetzt. Bis Februar 2004 wurde mit MSW Logo6.4h¹ in einer deutschen Übersetzung gearbeitet, da eine natürliche Formulierung von Problemlösungen im Vordergrund stand und diese nur in der Muttersprache möglich ist. Vorteile bot außerdem die sehr einfach und intuitiv zu bedienende Benutzeroberfläche (vgl. Abbildung 3.10).

Im Menü wurden nur die Befehle *Datei* → *Bewahre*, *Datei* → *Lade* und *Datei* → *Ade* zum Speichern und Laden von Logo-Programmen und dem Verlassen von MSW Logo

¹<http://www.ph-ludwigsburg.de/logo.html> (Abruf 03.08.05)

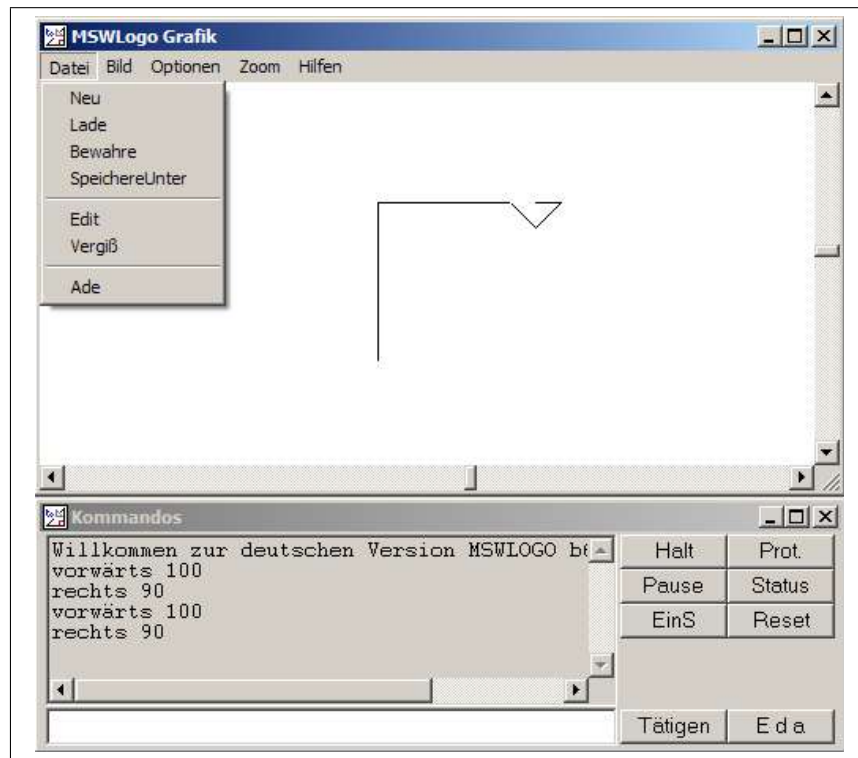


Abbildung 3.10: Benutzeroberfläche von MSW Logo

genutzt (vgl. Abbildung 3.11). Jede weitere Funktionalität musste aus der Kommandozeile des Hauptfensters aufgerufen werden. Eine einzelne Prozedur wurde mit dem Befehl `EDIT <Prozedurname>` und alle im Programm definierten Prozeduren mit `EDA` in den Editor geladen, um dort bearbeitet zu werden. Zum Schließen des Editors musste der Menüpunkt *Datei* → *Lerne u. Ende* aus der in Abbildung 3.11 dargestellten Menüstruktur ausgewählt werden. Dies geschah sehr intuitiv, da der Editor in einer Metapher als Lernort des Igels bzw. des Systems eingeführt worden war.

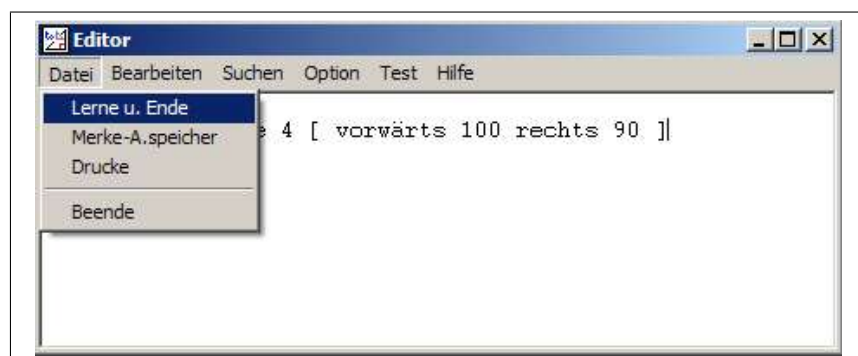


Abbildung 3.11: Benutzeroberfläche des Editors von MSW Logo

In der zweiten Phase des Unterrichtsversuchs wurde der Fokus stärker auf das *Programmieren im Großen* gelegt, so dass die Schülerinnen und Schüler an einem System arbeiten sollten, das die informatischen Grundkonzepte der *Objektorientierung*, *Modularisierung*, *Hierarchisierung* und *Modellierung* von Benutzerschnittstellen verdeutlichte. Diesen Ansprüchen wurde MSW Logo nicht gerecht. Aus dem weiteren Wunsch, mit einem neuen System auch ästhetisch ansprechende interaktive multimediale Projekte erstellen zu können, ergaben sich folgende Anforderungen:

- Sichtbarkeit objektorientierter Konzepte an der Programmoberfläche und im Programmcode
- Darstellung der Modulstruktur und ihrer hierarchischen Abhängigkeiten
- Möglichkeit zur einfachen Erstellung von Benutzerschnittstellen
- Einbindung multimedialer Objekte ohne Programmierung.

Logo-Systeme in deutscher Sprache, die nur annähernd diesen Anforderungen genügen, existierten zu Beginn des Unterrichtsversuchs nicht. Aus diesem Grund musste auf ein englischsprachiges System zurückgegriffen werden. Die Entscheidung fiel auf Imagine Logo².



Abbildung 3.12: Objekthierarchie und Kontextmenü in Imagine Logo

Sein objektorientierter Aufbau spiegelt sich an der Oberfläche in der Objekthierarchie der Entwicklungsumgebung, in den Kontextmenüs der einzelnen Bildschirmobjekte (vgl. Abbildung 3.12) und der Möglichkeit zur Änderung von Attributwerten in Registerkarten, wie in Abbildung 3.13 dargestellt, wider. Analoges gilt für die Ebene der Programmierung. Anstatt Kontextmenüs mit der rechten Maustaste auszuwählen, müssen Objekte im Programmcode unter ihrem Namen angesprochen und mit Methoden wie `setPos` oder `setBGColour` manipuliert werden.

In Imagine Logo können eigene Module durch Einblendung der Prozedurhierarchie (vgl. Abbildung 3.14) angezeigt werden. In ihr sind die entsprechenden Prozeduren linear angeordnet. Die Abhängigkeiten der Module untereinander werden jedoch nicht dargestellt.

²Logotron Educational Software, <http://www.logo.com/imagine> (Abruf 03.08.05)

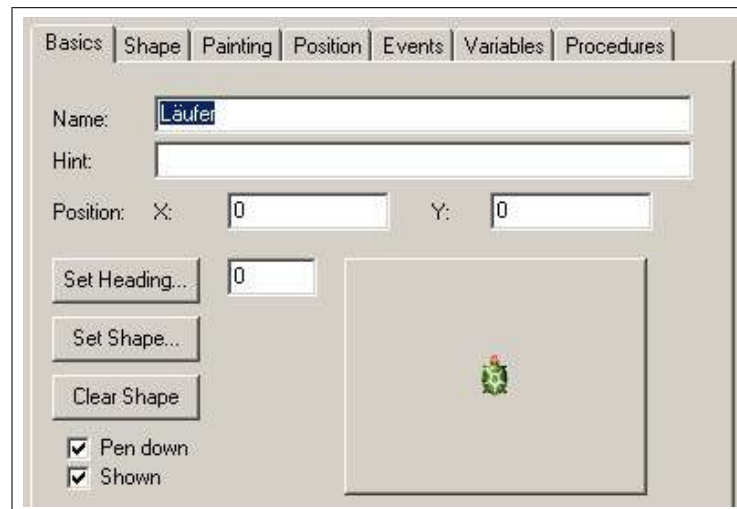


Abbildung 3.13: Registerkarte der Attribute eines Igel-Objekts in Imagine Logo

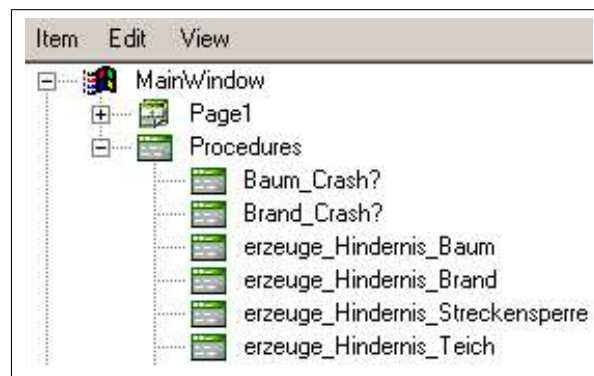


Abbildung 3.14: Prozedurhierarchie in Imagine Logo

Die Benutzerschnittstelle kann sehr einfach durch die Auswahl von Oberflächenobjekten aus der Symbolleiste erstellt werden. Es stehen, wie in Abbildung 3.15 angedeutet, *Ebenen*, *Schaltflächen*, *Textfelder*, *Schieberegler*, *Eingabeboxen*, *Verweismöglichkeiten ins Internet* und *Seiten* zur Verfügung. Nach der Auswahl eines Oberflächenobjekts kann dieses mit Hilfe der Maus an einer beliebigen Position im Arbeitsbereich eingefügt werden. Seine Attributwerte sind dann initialisiert und müssen gegebenenfalls über die **Change**-Methode im Kontextmenü angepasst werden. Genauso wie die Oberflächenelemente können multimediale Objekte eingebunden werden. Hier werden in der Symbolleiste Bilder-, Audio-, Video-, Musik- (siehe Abbildung 3.16) oder Web-Browserobjekte angeboten.

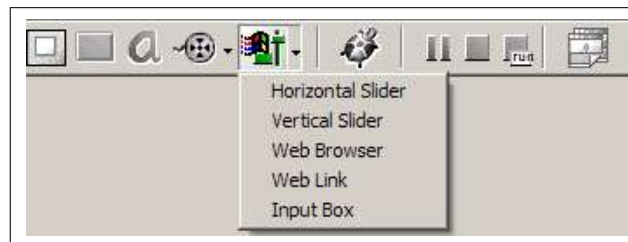


Abbildung 3.15: Oberflächenobjekte in Imagine Logo



Abbildung 3.16: Multimediaobjekte in Imagine Logo

3.3 Fachinhalte und informatische Grundkonzepte

Im Schuljahr 2003/2004 war in Baden-Württemberg an Realschulen der Bildungsplan von 1994 (Kultusministerium Baden-Württemberg, 1994b) gültig. Aus diesem wurden die Inhalte der Fächer Mathematik, Englisch, Deutsch und Musik ausgewählt, in deren Kontext die informatischen Grundkonzepte sinnvoll vermittelt werden konnten. Eine Übertragung auf die Inhalte des neuen Bildungsplans (Kultusministerium Baden-Württemberg, 2004b) ist jedoch leicht möglich.

Wirkprinzipien von Informatiksystemen

In den Fächern Deutsch und Englisch spielen sprachliche Strukturen eine Rolle, die sich systematisieren lassen. Ihre Beschreibungen können genutzt werden, um *automatisch* Instanzen zu erzeugen. Ähnlich können abgeleitete Substantive gebildet werden. Sie werden aus einem Wortstamm und einer Endung zusammengesetzt. Die Hervorhebung der entsprechenden Endungen unterstützt die Fehlervermeidung bei der Groß-/Kleinschreibung.

Alle Unterrichtsthemen, in denen Regeln gelernt werden, bieten sich für eine Einführung von *Algorithmen* an. In Musik sind dies die Positionen der Halbtonschritte in Dur- und Moll-Tonleitern, in Englisch die Bildungsgesetze der Verlaufsform der Verben und in Mathematik die Umwandlung der römischen Zahlen in das Dezimalsystem. Algorithmen erfordern eine formalisierte und somit präzise Beschreibung der Regeln. Sie können in Form eines Computerprogramms dem Rechner beigebracht werden. Die Richtigkeit kann durch intelligentes Testen überprüft werden.

Sachverhalte können häufig auf ganz unterschiedliche Art und Weise dargestellt werden. Bei Erfassung aller relevanten Eigenschaften handelt es sich um äquivalente Repräsentation.

tionen, die verschieden *codiert* sind. Zahlen können zum Beispiel als römische Zahlen oder Dezimalzahlen und Noten im Fünfliniensystem oder als Balkendiagramme repräsentiert werden.

Informatische Modellierung

In Sprachen wird zwischen Wortarten (z.B. Substantiv, Verb) und Satzgliedern (Subjekt, Prädikat, Genitivobjekt etc.) unterschieden. Im ersten Fall handelt es sich um einzelne Wörter, im zweiten um Wortgruppen wie „das Haus“ oder „dessen schöner Garten“, die als Einheit betrachtet werden. Ihre Darstellung erfordert unterschiedliche *Datenstrukturen*, da die Wörter der Satzglieder zusammengefasst werden müssen. Gleiches gilt in der Musik für die Unterscheidung zwischen einzelnen Noten und Melodiebausteinen, die aus mehreren Noten bestehen.

Modelle beschreiben die Realität abstrakt und beschränken sich auf die für einen Kontext wesentlichen Eigenschaften. Sie spielen im Fremdsprachenunterricht eine Rolle. Dort steht im Anfangsunterricht ein sehr begrenzter Wortschatz zur Verfügung, so dass nur mit einem Modell der Sprache gearbeitet wird. Beim Lernen grammatikalischer Regeln wird dies besonders deutlich, da zunächst nur diejenigen Regeln und Ausnahmen gelernt werden müssen, die für den gültigen Wortschatz relevant sind. Mit der Erweiterung des Wortschatzes erfolgt dann auch eine weitere Differenzierung der Grammatikregeln.

Als Einführung in die *objektorientierten Konzepte* der *Klassen*, *Attribute* und *Objekte* bieten sich in Musik Töne an. Leicht können Eigenschaften identifiziert und eine Klasse Ton mit Attributen Tonhöhe und Tonlänge definiert werden. Die eindeutige Namensgebung und der Aufruf von Methoden in der Objektorientierung kann bei der Animation von Objekten geübt werden.

Arbeitstechniken

Um das Konzept der *Modularisierung* anwenden zu können, muss eine Problemstellung komplex genug sein. In Mathematik bietet sich dafür die Erstellung von Bandornamenten und in den Sprachen die Generierung von Sätzen an. Im Zusammenhang mit der Modularisierung kann auch die *Hierarchisierung* thematisiert werden, da die Lösungen der Teilprobleme hierarchisch zur Gesamtlösung zusammengesetzt werden.

Das Prinzip der *Abstraktion* wird anhand der ebenen Figuren in Mathematik anschaulich. Die Begriffe *Viereck*, *Quadrat*, *Raute*, etc. definieren jeweils Klassen von unendlich vielen geometrischen Objekten, die sich in ganz speziellen Eigenschaften unterscheiden. Beim Quadrat zum Beispiel wird von der Seitenlänge abstrahiert. Gleiches gilt für Satzglieder oder Wortgruppen in Sprachen oder Töne in Musik.

Wenn die Orientierungsbefehle **VORWÄRTS** und **RECHTS** weggelassen und nur die Befehle zur Bewegung des Igels betrachtet werden, so ist die Formel zur Berechnung des Umfangs von Quadraten und Rechtecken dem *iterativen* Programm, das das Zeichnen dieser Figuren mit Hilfe einer Schleife realisiert, sehr ähnlich.


```
WIEDERHOLE 4 [ VORWÄRTS :Länge  RECHTS 90 ]  
U = 4 * Länge  
WIEDERHOLE 2 [ VORWÄRTS :Länge  RECHTS 90  VORWÄRTS :Breite  RECHTS 90 ]  
U = 2 * Länge * Breite
```

Die *Rekursion* bietet als elegante und äußerst kompakte Beschreibungsform Anknüpfungspunkte an das Fach Deutsch.

Informatische Beschreibungsmittel

Die standardisierten informatischen Beschreibungsmittel eignen sich zur Veranschaulichung der Fachinhalte ganz unterschiedlich. Zur Darstellung von Prozessen sind *Programmablaufpläne* sehr nützlich. Das Vorgehen bei der Umwandlung römischer Zahlen in Dezimalzahlen kann mit ihnen visualisiert werden. *Syntaxdiagramme* beschreiben statische Strukturen und werden zur formalen Definition einer Grammatik eingesetzt. Zur Veranschaulichung der verschiedenen Beziehungen zwischen den ebenen Figuren Viereck, Parallelogramm, Quadrat, Rechteck und Raute und der Vererbung ihrer Eigenschaften bieten sich *UML-Klassendiagramme* an. *Hierarchien* und *Netzwerke* spielen in fast allen Fächern zur Strukturierung fachlicher Zusammenhänge eine Rolle.

Fachnahe Inhalte

Die fachnahen Inhalte können den entsprechenden Unterrichtseinheiten aus dem Lehrplan zugeordnet werden. *Grammatik*, *Syntax* und *Semantik* sind Themen in den Fächern Englisch und Deutsch und *Funktionen* werden propädeutisch für die Klassenstufe 8 in Mathematik eingeführt.

Das Schema des Unterrichtskonzepts zur Vermittlung der informatischen Grundkonzepte im fächerübergreifenden Unterricht (vgl. Abbildung 2.5) kann somit noch um Logo und die vier Fächer erweitert werden. In dem daraus resultierenden Gesamtschema in Abbildung 3.17 werden die wichtigsten Komponenten und Interdependenzen beschrieben.

3.4 Unterrichtsmethoden

Die Anforderungen an die Schülerinnen und Schüler waren im Unterrichtsversuch aufgrund der unterschiedlichen Lerninhalte sehr hoch. Sie mussten informatische Grundkonzepte, Fachinhalte, die Programmiersprache Logo, informationstechnische Grundkonzepte und Schlüsselqualifikationen wie Selbstständigkeit, Gestaltungsfähigkeit etc. erlernen. Außerdem wurden die verschiedenen Inhalte dieser Gebiete immer wieder miteinander verknüpft. Damit sie durch den Grad der Komplexität nicht überfordert wurden, war es vor allem bei der Einführung neuer informatischer Grundkonzepte notwendig, die Instruktionsphasen nicht zu knapp zu gestalten. Das Konzept des *problemorientierten Lernens* eignete sich für das Erlernen der informatischen Grundkonzepte vor allem in

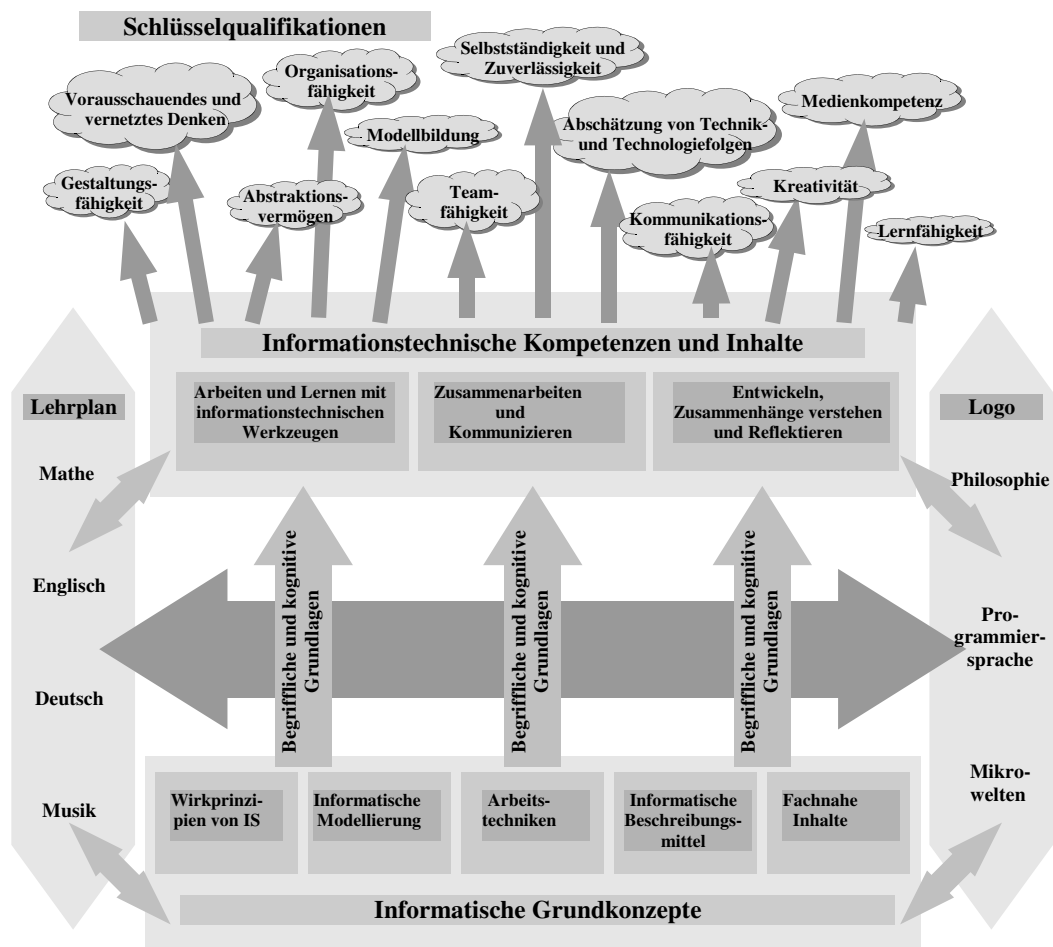


Abbildung 3.17: Gesamtkonzept

den Unterrichtseinheiten in der ersten Hälfte des Schuljahrs. Dadurch konnte eine angemessene Balance zwischen Konstruktion und Instruktion, wie sie von Schnurer & al. (2003, S. 156) gefordert wird, erreicht und die Vorteile beider Methoden vereint werden.

Beim problemorientierten Lernen wurden die authentischen Problemstellungen jeweils aus einem der vier Fächer gewählt, so dass sie eine unterrichtliche Relevanz für die Schülerinnen und Schüler hatten und das Erlernen der Fachinhalte in den anderen Unterrichtsstunden unterstützten. Viele informatische Grundkonzepte wurden auch in unterschiedliche fachliche Kontexte eingebettet. Dies hatte zur Auswirkung, dass die Fachinhalte untereinander stärker vernetzt wurden und die Konzepte flexibel anwendbar waren. Das selbstgesteuerte und soziale Lernen fand vor allem am Computer statt. Dort hatten alle die Möglichkeit, in einem eigenen Lerntempo mit dem Partner zu arbeiten, Lösungen mit ihm zu diskutieren, Alternativen auszuprobieren, sich die Arbeiten der anderen an den Rechnern anzuschauen oder sich Hilfestellungen bei den Mitschülerin-

nen und Mitschülern oder der Lehrerin zu holen. Theoretische Inhalte wie zum Beispiel die informatischen Beschreibungsmittel, deren Syntax und Semantik nicht explorativ erarbeitet werden können, und programmiersprachliche Konstrukte wurden dagegen instruktiv im Frontalunterricht eingeführt. Dieses Vorgehen verschaffte den Lernenden einen schnellen Überblick und unterstützte sie vor allem bei anfänglichen Programmierschwierigkeiten. Somit wurden die Vorteile der Instruktion, wie sie in Hense & al. (2001, S. 7) beschrieben sind, mit denen eines aktiven und konstruktiven Vorgehens optimal verknüpft und der Aussage von Mayer (2004, S. 18) Rechnung getragen, dass Instruktion auch in konstruktiven Lernansätzen ihre Bedeutung hat:

„Pure discovery did not work in the 1960s, it did not work in the 1970s, and it did not work in the 1980s, so after these three strikes, there is little reason to believe that pure discovery will somehow work today.“

Die Vermittlung informatischer Grundkonzepte in Mathematik, Englisch, Deutsch und Musik legte einen fächerübergreifenden Ansatz der Unterrichtskonzeption sehr nahe. Huber (1998) sieht als Vorteil des fächerübergreifenden Unterrichts seinen ganzheitlichen Ansatz, der eher der natürlichen Neugier des Kindes entgegenkommt, seine Problemorientierung durch die Arbeit an möglichst relevanten praktischen Problemen, seine Förderung reflexiven Denkens und sein Umgang mit der Heterogenität, die Vielfalt hervorbringen soll. Huber (1995) unterscheidet fünf verschiedene Formen fächerübergreifenden Unterrichts bezüglich ihrer Organisationsformen:

- Der *fächerüberschreitende Unterricht* findet in einem einzelnen Fach statt. Von diesem ausgehend werden die Fachgrenzen überschritten und Inhalte anderer Fächer einbezogen.
- Im *fächerverknüpfenden Unterricht* steht ein zentrales Thema im Mittelpunkt, das mehrere Fächer miteinander verbindet.
- *Fächerkoordinierender Unterricht* ist ein fächerverknüpfender Unterricht, in dem die beteiligten Fächer sehr stark kooperieren.
- Im *fächerergänzenden Unterricht* wird die Fächerordnung zugunsten von themen- und projektorientiertem Unterricht ausgesetzt. Er ergänzt den Fachunterricht.
- Der *fächeraussetzende Unterricht* hebt für einen festgelegten Zeitraum den gesamten Fachunterricht, wie zum Beispiel in Projektwochen, auf.

Das Unterrichtskonzept beinhaltete verschiedene Organisationsformen. Häufig fand ein fächerkoordinierender Unterricht statt, da die informatischen Grundkonzepte im Kontext mehrerer Fächer angesprochen wurden. Es bot sich zum Beispiel an, das Thema Abstraktion sowohl im Zusammenhang mit den ebenen Figuren in Mathematik als auch der Satzglieder in den Sprachen zu thematisieren.

Der Lehrplan der Klassenstufe 5 enthielt auch Fachinhalte, deren Erarbeitung ein fächer-

verknüpfendes Vorgehen anbot. So wurde die Einführung der Tonlänge in Musik mit Konzepten der Bruchrechnung verknüpft. Durch die Analyse der Sprachstrukturen in Deutsch und Englisch ergaben sich Anknüpfungspunkte. Diese wurden durch die Verwendung von Syntaxdiagrammen zur Darstellung der Satzstrukturen sehr deutlich. Auch Beschreibungen waren nicht nur Thema des Fachs Deutsch. Sie konnten in Musik bei der Charakterisierung von Melodiebausteinen angewendet werden. Alle Beispiele verdeutlichen, dass ausgehend von einem Fach die Fächergrenzen überschritten und die Bezüge zu anderen Inhalten hergestellt wurden.

Die mehrwöchige Projektarbeit wurde als fächerergänzender Unterricht konzipiert, da in den vorgegebenen Pflichtteilen weiterhin an Fachinhalten aus Mathematik, Englisch, Deutsch und Musik gearbeitet wurde. Nur während einer kompakten Methodikwoche war der Fachunterricht komplett ausgesetzt.

In Moegling (1998) werden eine Reihe von Methoden für fächerübergreifendes Lernen beschrieben. Im Unterrichtskonzept zur Vermittlung der informatischen Grundkonzepte wurde meistens das exemplarische Lernen als didaktisch-methodischer Weg gewählt. Im Rahmen eines speziellen Fachthemas wurden die informatischen Grundkonzepte als ganz allgemeine Prinzipien, Strukturen und Konzepte eingeführt und geübt. In den Fachkontexten hatten sie die Rolle eines Denk- oder Hilfsmittels. Ihre allgemeine Bedeutung wurde durch Anwendung in weiteren Fächern hervorgehoben. Im fächerergänzenden Unterricht wurde die Projektmethode angewendet.

3.5 Vorbereitung der Fachlehrerinnen

Ein sehr wichtiger Erfolgsfaktor für den Unterrichtsversuch war die Einstellung der beteiligten Fachlehrerinnen. Sie hatten letztendlich die Verantwortung für die Unterrichtsinhalte in den Fächern und insgesamt eine Stunde weniger Zeit, sie zu vermitteln. Zunächst konnte von einer intrinsisch hohen Motivation ausgegangen werden, da sie sich freiwillig zur Mitarbeit im Forschungsprojekt bereit erklärt hatten. Trotzdem sollten sie schon einige Monate vor Schuljahresbeginn das Konzept zur Vermittlung der informatischen Grundkonzepte genauer kennen lernen. In mehreren Sitzungen (vgl. Tabelle 3.1) arbeiteten sie ähnlich wie später die Schülerinnen und Schüler mit Logo an einzelnen Unterrichtsthemen. Jeder Lehrerin stand ein Computer zur Verfügung. Durch eigene Lernerfahrungen wurde ihnen sehr schnell deutlich, dass es sich bei dem Konzept nicht um den sonst üblichen Unterricht zur *Informationstechnischen Grundbildung* handelte, sondern das Erlernen der informatischen Grundkonzepte im Mittelpunkt stand und mit Hilfe der Logo-Programmierung vertieft und angewendet wurde. Außerdem erfuhren sie vorab, in welcher Form die informatischen Konzepte in die Inhalte ihrer Fächer integriert wurden. Obwohl sich die Fähigkeiten der Lehrerinnen im Umgang mit dem Computer deutlich unterschieden, konnten sie die informatischen Grundkonzepte und

Logo-Programmierung in der Weiterbildung gemeinsam erlernen. Die Mathematiklehrerin, die auch das Fach *Mensch und Umwelt* und Biologie unterrichtete und das dritte Jahr nach ihrem Referendariat an der Schule tätig war, nutzte den Computer intensiv als Werkzeug zur Erstellung von Unterrichtsmaterialien, zur Verwaltung von Schülerdaten, Recherchen im Internet und zum Schreiben von E-Mails. Ihre Kompetenzen hatte sie bereits im Studium erworben. Die berufserfahrene Englischlehrerin mit Zweitfach Sport, die bis zum Zeitpunkt des Unterrichtsversuchs die eigene Arbeit am Computer möglichst vermied, hatte sehr wenige Vorkenntnisse. Die Deutsch- und Musiklehrerin, die bereits über 20 Jahre im Schuldienst war, führte Recherchen im Internet durch und schrieb E-Mails und einfache Textdokumente.

Termin	Thema	Anwendungen
10.04.2003 (16:30-18:30 Uhr)	Vorstellung der Gesamtkonzeption des Unterrichtsversuchs	
14.04.2003 (10:00-12:00 Uhr)	Befehle aufrufen, Befehlsfolgen, Wiederholung, Befehle erstellen	Igel-Grafik: Quadrat, Rechteck
05.05.2003 (19:00-21:00 Uhr)	Befehle mit Eingabeparametern erstellen	Igel-Grafik: Quadrat, Quadratfiguren, Drehfiguren, Bandornamente, Spiegelungen, Blumen
19.05.2003 (19:00-21:00 Uhr) Engischlehrerin	Funktionen, Maschinendarstellung, Funktionsgeflechte, Konstruktoren und Selektoren für Wörter	Spielereien mit Wörtern
26.05.2003 (19:00-21:00 Uhr)	Funktionen, Maschinendarstellung, Funktionsgeflechte, Konstruktoren und Selektoren für Wörter	Spielereien mit Wörtern
02.06.2003 (19:00-21:00 Uhr)	Funktionen, Konstruktoren und Selektoren für Sätze	Spielereien mit Sätzen, Phrasendreschmaschine
06.06.2003 (10:00-12:00 Uhr) Englisch-/Deutschlehrerin	Bedingungen, Unterscheidung von Variablennamen, Prozedurnamen und Wörtern, Unterscheidung von Befehlen und Funktionen	Unbestimmter englischer Artikel, Plural englischer Substantive
16.06.2003 (19:00-21:00 Uhr)	Einführung der endständigen Rekursion, Befehle	Igel-Grafik
14.07.2003 (19:00-21:00 Uhr)	Fortsetzung der endständigen Rekursion, Funktionen	Wort- und Satzspielereien
25.07.2003 (10:00-12:00 Uhr)	Einführung in Musik, Listen, Codierung	Eigene Melodien, Transponieren, Tonleiter
02.09.2003 (19:00-21:00 Uhr)	Wiederholung: Befehle, Funktionen, Bedingungen, Variablen, Variablenwerte	Bandornamente, abgeleitete Substantive, Perfekt und Partizip Perfekt englischer Verben, Wörterbuch

Tabelle 3.1: Logo-Fortbildung der Lehrerinnen

3.6 Vorbereitung der Eltern

Auch die positive Einstellung der Eltern zum Unterrichtsversuch hatte eine große Bedeutung für seinen Erfolg. Sie sollten nicht das Gefühl haben, dass ihre Kinder durch die zusätzlichen Anforderungen der Vermittlung der informatischen Grundkonzepte, dem Arbeiten am Computer und dem Programmieren in Logo zu sehr belastet werden. Für die meisten war der Schulwechsel mit einem Ortswechsel verbunden. Der Schulweg wurde dadurch mit einem öffentlichen Bus zurückgelegt. Die Kinder kannten sich zum großen Teil nicht, so dass alle erst einmal ihren Platz innerhalb der Klasse finden mussten. Neu war auch der Fachunterricht, der von verschiedenen Lehrerinnen und Lehrern durchgeführt wurde. In jeder Schulstunde mussten sie sich auf eine andere Person einstellen.

Gleich zu Beginn des Schuljahrs wurde ein Fortbildungsangebot an vier Abenden eingerichtet, an dem die Eltern selbst ausprobieren konnten, in welcher Form ihre Kinder über das Schuljahr hinweg mit Logo arbeiteten. Gleichzeitig lernten sie die Logo-Philosophie, den Nutzen informatischer Grundkonzepte, die Zusammenhänge zwischen den informatischen Grundkonzepten und den Fachinhalten und das Gesamtkonzept in einem Überblick kennen. Etwa die Hälfte der Eltern nahm das Angebot an. Die Inhalte der vier Veranstaltungen sind in Tabelle 3.2 abgebildet.

Termin	Thema	Anwendungen
30.09.2003 (19-21 Uhr)	Befehle aufrufen, Befehlsfolgen, Wiederholung, Befehle erstellen	Ebene Figuren, Verschiebungen, Drehfiguren
14.10.2003 (19-21 Uhr)	Befehle mit Eingabeparametern erstellen, Listen, Funktionen, Maschinendarstellung, Funktionsgeflechte	Quadrat- und Rechteckfiguren, Glücksrad, Spielereien mit Wörtern, englische Kurzsätze
21.10.2003 (19-21 Uhr)	Befehle mit Eingabeparametern erstellen, Funktionen, Konstruktoren und Selektoren für Wörter, Bedingungen	Spinnennetz, Spielereien mit Wörtern, Imperativ, Komparativ, Verben ohne Vorsilben, unbestimmter englischer Artikel, Plural englischer Substantive
28.10.2003 (19-21 Uhr)	Einführung in die Musik, Listen, Befehle, Funktionen, Codierung	Zahlenrätsel, Tonleitern, Rhythmen, Transposition, Krebs, Umkehrung

Tabelle 3.2: Logo-Fortbildung der Eltern

4 Unterrichtseinheiten zu informatischen Grundkonzepten

Der Unterrichtsversuch gliedert sich in drei Phasen, die sich bezüglich der Methoden, der Wissensvermittlung und -anwendung und der Integration der informatischen Grundkonzepte in die Fachinhalte unterscheiden:

1. Vermittlung einzelner informatischer Grundkonzepte in einem Fach
2. Anwendung informatischer Grundkonzepte im fächerübergreifenden Projektunterricht
3. Erarbeitung informatischer Grundkonzepte anhand fachunabhängiger Themen.

In der ersten Phase bis Ende März war der instruktionale Anteil des Unterrichts am größten, da informatische Grundkonzepte und Logo-Sprachstrukturen eingeführt wurden. Die Wissensvermittlung stand in kürzeren Unterrichtseinheiten im Zusammenhang eines Fachinhalts im Vordergrund. Anschließend wurden die erlernten informatischen Grundkonzepte in einer fünfwöchigen Projektarbeit bei der Erstellung einer multimedialen Rittergeschichte angewendet. Innerhalb des Projekts wurde gleichzeitig an Inhalten der Fächer Mathematik, Englisch, Deutsch und Musik gearbeitet. In der dritten Phase wurden schließlich weitere informatische Grundkonzepte in einem meist fachunabhängigen Unterricht erarbeitet und vertieft.

Die einzelnen Lehr-/Lernarrangements dieser drei Phasen werden in den folgenden Kapiteln kompakt dargestellt. Zu Beginn steht jeweils eine tabellarische Auflistung der informatischen Schwerpunkte, Fachinhalte und verwendeten Logo-Mikrowelten. Es schließt sich eine ausführliche Beschreibung der Einbettung der informatischen Grundkonzepte in die Fachinhalte, der didaktischen Überlegungen zur Vermittlung der informatischen Grundkonzepte und der Erprobung der Unterrichtseinheiten an.

4.1 Fachunterricht

Die Reihenfolge der im Folgenden vorgestellten Unterrichtseinheiten entspricht ihrer zeitlichen Abfolge während des Schuljahrs. Die vier Wochenstunden zur Vermittlung der informatischen Grundkonzepte wurden sehr flexibel auf die Inhalte der Fächer verteilt. Die einzelnen Einheiten waren auf zwei bis vier Stunden angelegt. Meistens wurde an zwei Themen parallel gearbeitet.

Neue informatische Grundkonzepte		
Fachinhalt	Mikrowelt	Wiederholung
Befehle, Befehlsfolgen		
Mathematik: Berechnung einer Streckenlänge auf einem Stadtplan	Stadt-Mikrowelt	Metapher <i>Igel spielen</i>
Mathematik: Quadrat, senkrechte Geraden	Igel-Mikrowelt	Algorithmus, Planung einer <i>komplexen</i> Igelbewegung
Iteration		
Mathematik: Quadrat, Rechteck	Igel-Mikrowelt	Kompakte Beschreibung, Kontrollstruktur
Repräsentationsformen		
Musik: Tonleiter	Musik-Mikrowelt	Datentyp Liste
Befehle definieren		
Mathematik: Quadrat, Rechteck, Parallelogramm, Raute	Igel-Mikrowelt	Spracherweiterung, systematische Beschreibung von Strukturen, Abstraktion, Prozeduren, Automatisierung
Fehlende Intelligenz in Computerprogrammen		
Englisch: Vokabeltrainingsprogramm für die erste Schulbucheinheit	Vokabeltraining-Mikrowelt	Intelligenz von Computersystemen, Begrenztheit
Befehle mit Parametern		
Mathematik: Quadrat, Quadratfigur, Rechteck, Rechteckfigur	Igel-Mikrowelt	Abstraktion, Module, gebundene Variablen
Funktionen		
Deutsch: Abgeleitete Substantive, Tageseit, Großschreibung	Wort-Mikrowelt, Drucke-Mikrowelt	Metapher <i>Glücksrad</i> , Datentyp Liste, Automatisierung, Syntax, Semantik
Englisch: Englische Sprachdialoge	Dialog-Mikrowelt	Begrenztheit, Formalisierung, Digitalisierung
UML-Klassendiagramme		
Mathematik: Beziehungen zwischen ebenen Figuren		Oberklasse, Unterklasse, Eigenschaften, Vererbung
Modularisierung		
Musik: Musikbausteine anhören, beschreiben und zusammensetzen	Melodie-Mikrowelt	Bottom-up
Funktionsgeflecht		
Englisch: Struktur von „This is“-Sätzen und „Is“-Fragen	Satz-Mikrowelt, Drucke-Mikrowelt	Maschinen, Abstraktion, zusammengesetzte Maschinen, Datenstruktur geschachtelter Listen, Modularisierung

Neue informatische Grundkonzepte		
Fachinhalt	Mikrowelt	Wiederholung
Algorithmus		
Mathematik: Umwandlung römischer Zahlen ins Zehnersystem	Römische Zahlen-Mikrowelt	Regelwerk mit Ausnahmen lesen, schriftlicher Algorithmus und Programmablaufplan lesen, Bedingungen
Äquivalenz von Repräsentationsformen		
Musik: Musikbausteine anhören, beschreiben und Repräsentationen im Fünfliniensystem Balkendiagrammen zuordnen	Papageno-Mikrowelt	Module
Abstraktion		
Englisch: „Is“-Fragen	Satz-Mikrowelt, Drucke-Mikrowelt	Befehle, Funktionen, Funktionsgeflechte, zusammengesetzte Maschinen, Modularisierung, Abstraktion
Programmablaufpläne erstellen		
Deutsch: Steigerung von Adjektiven	Wort-Mikrowelt, Drucke-Mikrowelt	Funktionen, Maschinen, Formulierung von Regeln, Bedingungen
Automatisierung		
Allgemein: Fadenbilder zweier Igel	Fadenbilder-Mikrowelt	Objekt, Planung <i>komplexer</i> Bilder, Modularisierung
Allgemein: Weihnachtskarten (Englisch: Kurztex-te, Mathematik: Drehfiguren, Verschiebung)	Weihnachtskarten-Mikrowelt	Module, Planung <i>komplexer</i> Bilder
Fehleranfälligkeit von Computersystemen		
Mathematik: Zweisatz		
Syntaxdiagramme		
Deutsch: Satzlehre, Satzglieder	Satz-Mikrowelt, Drucke-Mikrowelt	Maschinen, Funktionen, Datenstrukturen, Abstraktion, Modularisierung, Syntax, Semantik
Automatisierte Tests		
Englisch: Unbestimmter englischer Artikel	Wort-Mikrowelt, Satz-Mikrowelt, Drucke-Mikrowelt	Schriftliche Formulierung von Regeln, Algorithmus, Programmablaufplan erstellen, Maschinen, Funktionen, Funktionsgeflechte, zusammengesetzte Maschinen, Bedingungen, Umformung formaler Beschreibungsformen

Neue informatische Grundkonzepte		
Fachinhalt	Mikrowelt	Wiederholung
Objekte in unterschiedlichen Repräsentationsformen und ihre Umwandlung		
Musik: Fünfliniensystem, Takt, Taktstrich, Notenschlüssel	Musik-Mikrowelt	Attribute, Funktionen
Grammatik		
Englisch: Satzlehre, Sätze mit s-Genitiv		Syntaxdiagramme
Erweiterung von Repräsentationsformen bei Einführung neuer Attribute		
Musik: Tondauer, Notenhäse, Taktart, Grundschrift	Musik-Mikrowelt	Umwandlung in verschiedene Repräsentationsformen, schriftliche Formulierung von Regeln, Algorithmus, Datentypen, Noten als Objekte
Attribute		
Mathematik: Potenzen	Potenz-Mikrowelt	Maschinen, Funktionen, Syntaxdiagramme
Codierung		
Deutsch: Rechtschreibung durch Geheimcodes		<i>Komplexe</i> Programmablaufpläne lesen
Äquivalenz formaler Beschreibungsformen		
Englisch: Verlaufsform im Präsens	Wort-Mikrowelt, Satz-Mikrowelt, Drucke-Mikrowelt	Syntaxdiagramme, Maschinen, Funktionen, schriftliche Formulierung von Regeln, Algorithmus, Programmablaufplan, Funktionen
Hierarchische Darstellung mit Mindmaps		
Mathematik: Wiederholung Potenzen		
Objekte und Ereignisse an der Programmoberfläche		
Mathematik: Potenzlernprogramm	Potenzspiel-Mikrowelt	Klasse <i>Schaltfläche</i> und <i>Igel</i> , Ereignisse (<i>onClick</i> , <i>onPush</i>), Methoden
Hierarchie		
Deutsch: Rechtschreibung, Wörter mit <i>ie</i> und <i>ieh</i>	Pegasus-Mikrowelt	Baum, Blätter, Wurzel, Ebene, Attribute, Klasse <i>Textfeld</i> und <i>Seite</i>
Objektorientierte Programmierung		
Englisch: Eine kleine Geschichte	Jim-Mikrowelt	Namensgebung, Objekte, Methoden
Konstruktion eines Netzwerks		
Englisch: Beschreibung verwandtschaftlicher Verhältnisse		

Tabelle 4.1: Informatische Grundkonzepte eingebettet in Fachinhalte

Befehle, Befehlsfolgen

Mit der Igel-Grafik erfolgte für die Schülerinnen und Schüler ein sehr motivierender Einstieg in den Unterricht zur Vermittlung der informatischen Grundkonzepte und von Logo. Wie in der Literatur beschrieben (vgl. Shimabukuro 1988) kommandierte die Klasse zunächst einen Mitschüler mit verbundenen Augen durch den Klassenraum. Dabei erlernten die Kinder die Angabe von präzisen, nach einem vorgegebenen Schema gebildeten Befehlen und bauten eine Beziehung zwischen dem Arbeiten mit dem Igel und der eigenen Körpervorstellung auf. Genau wie sie selbst in der ersten Schulwoche die neue Schule und den Schulort erkundet hatten, mussten sie in ihrer ersten Aufgabe am Computer den Igel kreuz und quer durch eine fiktive Stadt (vgl. Abbildung 4.1) schicken und verschiedene Orte mit den Befehlen **VORWÄRTS 50**, **VORWÄRTS 100**, **VORWÄRTS 150**, **VORWÄRTS 200**, **RECHTS 90** und **LINKS 90** ansteuern lassen. Die mehrfache Angabe des **VORWÄRTS**-Befehls mit unterschiedlichen Längenangaben sollte die Kinder zum Experimentieren mit verschiedenen Aktualparametern auffordern. Ziel der Arbeit war, den kürzesten Weg, auf dem der Igel alle Orte der Stadt besuchen konnte, zu finden und diesen zu dokumentieren. Die Streckenlänge mussten sie selbst berechnen. Die Stadt-Mikrowelt konnte mit den Befehlen **start** und **Stadtplan** geladen werden, so dass bereits nach 30 Minuten Unterricht die ersten Logo-Befehle im System geschrieben wurden.

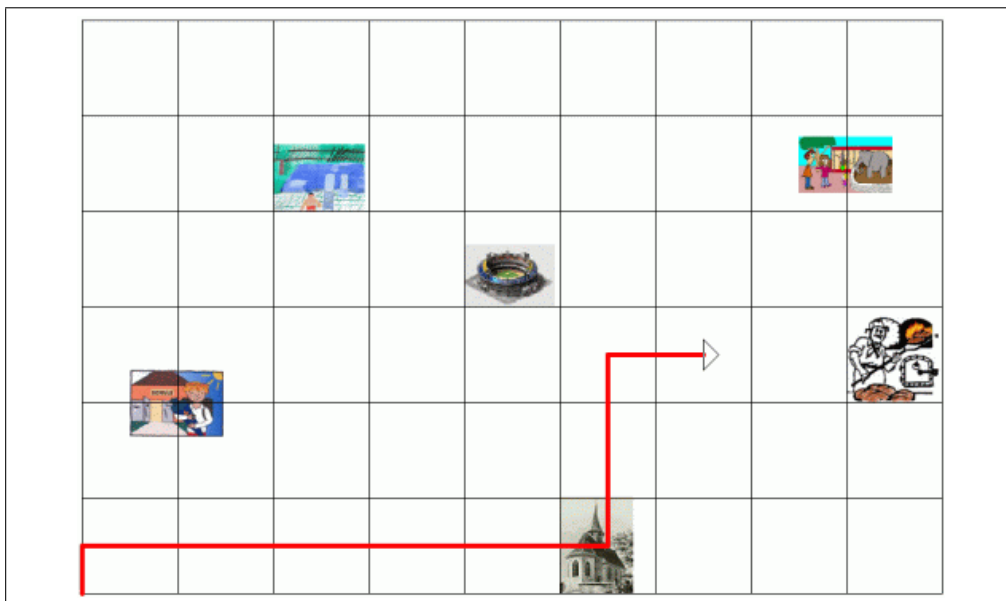


Abbildung 4.1: Stadt-Mikrowelt

In einer zweiten Aufgabe zeichneten die Kinder mit dem Igel ein Quadrat. Die Einführung erfolgte zunächst mit einem realen Holzigel im Klassenverband an der Tafel. Der Holzigel hatte wie in Logo eine spitze Nase und konnte sich nur in Richtung dieser fortbewegen. Außerdem hatte er in der Mitte ein Loch, in dem eine Kreide steckte. Beim Laufen

hinterließ er dadurch eine Spur. Gemeinsam erarbeitete die Klasse die für das Quadrat notwendigen Befehle. Jeder wurde einzeln notiert und der Holzigel entsprechend gedreht oder bewegt. Somit wurde der spätere Ablauf am Computer schrittweise simuliert. Anschließend präsentierte die Forscherin noch die Möglichkeit der Eingabe ganzer Befehlsfolgen, bevor die Schülerinnen und Schüler ihre Arbeit am Rechner begannen.

Die Mitarbeit in dieser ersten Unterrichtseinheit war sehr gut. In einer Art Wettkampf versuchten die Schülerinnen und Schüler hoch motiviert den kürzesten Weg durch die Stadt zu finden. Auch die gemeinsame Erarbeitung des Quadrats erfolgte unter Beteiligung fast aller. Die Umsetzung bereitete keinerlei Probleme, so dass alle Schüler Erfolgserlebnisse hatten. Einige experimentierten selbstständig mit sehr großen Zahlen.

Iteration

Die sehr mühsame Eingabe der jeweils acht Befehle für ein neues Quadrat motivierte eine kompaktere Beschreibung der Lösung. Im Unterrichtsgespräch fand die Klasse heraus, welche Teile der Lösung iteriert werden mussten und wie oft dies zu geschehen hatte. Eine Umsetzung in Logo erfolgte mit dem **WIEDERHOLE**-Befehl. Auf einem Hausaufgabenblatt vertieften die Schülerinnen und Schüler diesen Abstraktionsprozess von der schrittweisen Konstruktion einer Figur zur abstrakteren Formulierung ihrer Eigenschaften mit Hilfe von Iterationen anhand der ebenen Figur des Rechtecks. Sie mussten zunächst ein Rechteck mit Geodreieck und Bleistift zeichnen, die Eigenschaften der Figur schriftlich formulieren, dann die notwendigen Igel-Befehle notieren und schließlich die kompakte Darstellung mit dem **WIEDERHOLE**-Befehl entwickeln. Durch die Anordnung der Linien (vgl. Abbildung 4.2), auf denen die einzelnen Befehle der Lösung eingetragen werden mussten, wurde der Abstraktionsschritt von der sequentiellen zur iterativen Repräsentation unterstützt. Aus dieser Darstellung konnten die Anzahl der Wiederholungen und der wiederkehrende Teil sehr leicht abgelesen werden.

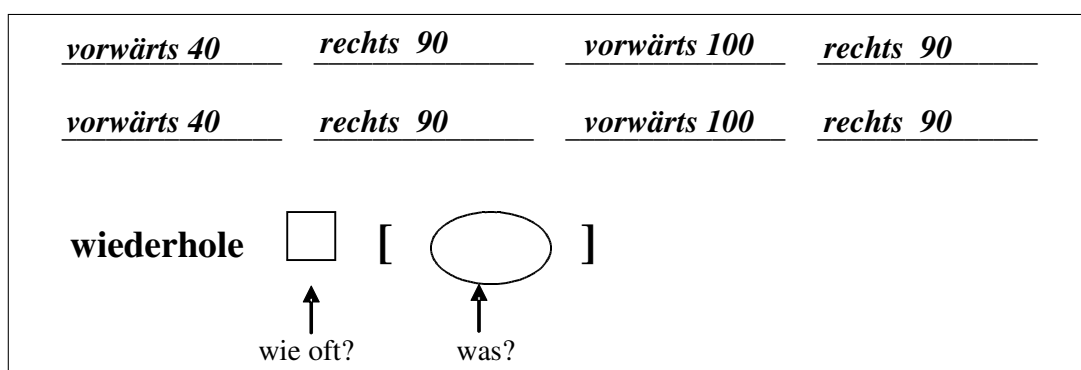


Abbildung 4.2: Abstraktion des Befehls Rechteck

Die Umsetzung in die formale Beschreibungsform der Programmiersprache Logo wurde durch die Beschriftung der Platzhalter des **WIEDERHOLE**-Befehls erleichtert.

WIEDERHOLE 2 [VORWÄRTS 40 RECHTS 90 VORWÄRTS 100 RECHTS 90]

Trotz der Besprechung der Hausaufgabe und der Implementierung des Rechtecks in Logo im Unterricht war die Ergebnissicherung in Form der Rechtecke auf den Arbeitsblättern nur in 18 Fällen korrekt. Im Unterricht gaben einige Schülerpaare weiterhin noch einzelne Befehle in das Logo-System ein und benutzten den WIEDERHOLE-Befehl nicht. Für sie war der Übergang von der sequentiellen zur iterativen Darstellung zu schnell. Andere hingegen spielten bereits mit Zahlen und erzeugten interessante Muster am Bildschirm. Eine Gruppe entwarf bereits Drehbilder.

Repräsentationsformen

Töne waren im Logo-System zunächst in Form von Zahlen repräsentiert. Tabelle 4.3 zeigt die Zuordnung der Zahlen zu den Tönen. Halbtonschritte waren als Einer- und Ganztonschritte als Zweierschritte implementiert.

Ton	Zahl
c^1	60
d^1	62
e^1	64
f^1	65
g^1	67
a^1	69
h^1	71
c^2	72

Abbildung 4.3: Repräsentation von Tönen als Zahlen

Die erste Tonleiter wurde als Rätsel eingeführt.

Aufgabe: Bilde nach folgenden Vorschriften Reihen

1. Die Reihe hat genau 8 Zahlen.
2. Die erste Zahl der Reihe ist die 55.
3. Die letzte Zahl der Reihe ist die 67.
4. Zweimal ist eine Zahl um 1 größer als ihre Vorgängerzahl.
5. Fünfmal ist eine Zahl um 2 größer als ihre Vorgängerzahl.

Die Beschreibung war bewusst nicht eindeutig gewählt. Die Schülerinnen und Schüler bekamen die Aufgabe, drei verschiedene Reihen zu bilden und diese sich mit dem Befehl `spiele` in der Musik-Mikrowelt anzuhören. Nach einer Experimentierphase wurden dann zwei Reihen, nämlich die Dur- und die natürliche Moll-Tonleiter, ausgezeichnet und ihr Aufbau genau definiert. Zur Vorbereitung des Algorithmusbegriffs wurden weitere Rätsel zu Tonleitern mit eindeutigen Beschreibungen als Zahlen- und Buchstabenrätsel bearbeitet. Eine Vertiefung erfolgte durch die Formulierung eigener Rätsel.

Schwierigkeiten bereitete die Unterscheidung von Befehlen und Parametern in Logo durch das Vergessen des notwendigen Leerzeichens. Eine Gruppe experimentierte bereits mit dem `spiele`-Befehl, erzeugte chromatische Tonleitern und setzte den `WIEDERHOLE`-Befehl ein, um Tonleitern mehrfach abzuspielen.

Befehle definieren

In Mathematik war mit dem `WIEDERHOLE`-Befehl eine verkürzte Notation für Quadrate und Rechtecke eingeführt worden. Durch die Definition eigener Befehle als Spracherweiterung lernten die Schülerinnen und Schüler als weitere Abstraktionsstufe den Prozedurbegriff kennen. Mit den neuen Befehlen `Quadrat` und `Rechteck` zeichnete der Igel die ebenen Figuren, ohne dass Aussagen über deren Definition gemacht werden mussten. Ein in der Frontalphase eingesetztes Arbeitsblatt zur Erklärung der Funktionsweise des Logo-Editors erwies sich als sehr hilfreich. Alle Kinder der Klasse konnten ohne größere Schwierigkeiten eigene Befehle erstellen.

Fehlende Intelligenz in Computerprogrammen

In der ersten Unterrichtseinheit in Englisch wurde den Schülerinnen und Schülern ein sehr einfaches Vokabellernprogramm zur Verfügung gestellt, mit dem sie ihre ersten englischen Vokabeln trainieren konnten.

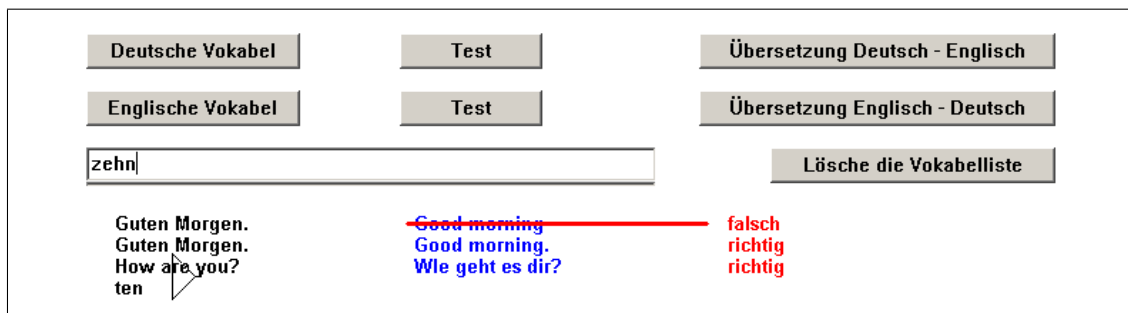


Abbildung 4.4: Benutzeroberfläche des englischen Vokabeltrainers

Intern waren die Vokabeln als deutsch-englische Paare gespeichert. Genau die exakte Paarkombination wurde mit dem Programm überprüft. Dies führte zum Beispiel in der ersten Zeile in Abbildung 4.4 dazu, dass das Programm die eigentlich richtige Übersetzung „Good morning“ von „Guten Morgen.“ ablehnte, da der abschließende Punkt fehlte. Die Kinder verloren also schon zu Beginn des Unterrichtsversuchs die Illusion, dass Computersysteme in der Lage sind, sich intelligent zu verhalten oder Handlungen von Menschen einschätzen zu können. Ihr Ärger über die eigentlich richtigen Antworten und das Herausfinden der Fehlerquelle erzeugten die notwendige Aufmerksamkeit.

Befehle mit Parametern

In einer weiteren Unterrichtseinheit wurde die Flexibilität des `Quadrat`-Befehls erhöht (vgl. Kapitel 3.1.2). Der Befehl aus einer vorherigen Stunde wurde an der Tafel notiert

und dann schrittweise für Quadrate unterschiedlicher Größen durch Ausstreichen und Ersetzen der Seitenlänge verbessert.

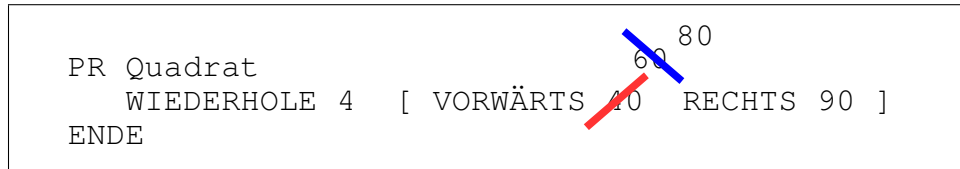


Abbildung 4.5: Befehl mit Parameter

Die Schülerinnen und Schüler erkannten, welcher Teil der Befehlsdefinition flexibel gestaltet werden musste. Dieser wurde durch einen Platzhalter in Form eines Dreiecks ersetzt.

```
PR Quadrat
  WIEDERHOLE 4 [ VORWÄRTS △ RECHTS 90 ]
ENDE
```

Da auf der Tastatur kein Dreieckssymbol zu finden war, sahen die Kinder leicht ein, dass sie statt des Dreiecks besser einen Namen wie *Länge* verwendeten. Der führende Doppelpunkt wurde mit der Unterscheidung von Namen und Platzhaltern motiviert. Entsprechend des Aufrufmusters **Quadrat 40** wurde schließlich noch der Platzhalter direkt hinter den Befehlsnamen geschrieben.

```
PR Quadrat :Länge
  WIEDERHOLE 4 [ VORWÄRTS :Länge RECHTS 90 ]
ENDE
```

Dieser noch abstraktere **Quadrat**-Befehl zum Zeichnen eines Quadrats beliebiger Größe wurde zur Erstellung der beiden Quadratfiguren in Abbildung 4.6 verwendet. In einer sehr guten Lösung wurde die linke Figur in einem eigenen Befehl gekapselt und zur Erzeugung der rechten Figur verwendet. Wie in der Auswertung in Kapitel 6.4.1 dargestellt wird, gelang dies jedoch nicht sehr vielen Schülerinnen und Schülern.

Funktionen definieren

Während des Unterrichtsversuchs wurde zur Generierung von Wörtern und Sätzen häufig die Funktion **ZEL** eingesetzt. Ähnlich wie beim Drehen eines Glücksrads gibt sie zufällig ein Element aus einer Liste zurück. In Abbildung 4.7 ist ein Glücksrad zur Auswahl eines Wochentags dargestellt, das in Logo als Funktion **Wochentag** folgendermaßen implementiert werden kann:

```
PR Wochentag
  RG ZEL [Montag Dienstag Mittwoch Donnerstag Freitag Samstag Sonntag]
ENDE
```

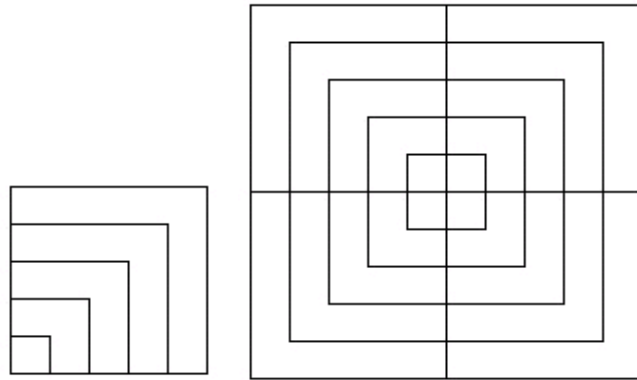


Abbildung 4.6: Quadratfiguren

Das Glücksrad bzw. die Funktion gibt den Wochentag zurück, der sich auf der Scheibe unten am Trichter befindet bzw. von der Funktion ZEL ausgewählt wird.

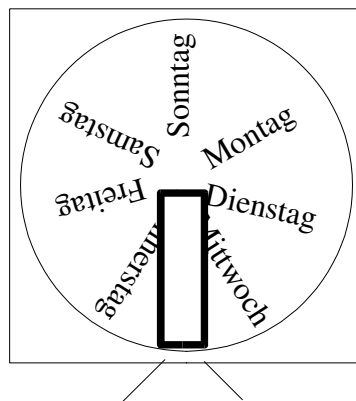


Abbildung 4.7: Metapher *Glücksrad*

Im Fach Deutsch wurde das Konzept der Funktionen am Beispiel von Zeitangaben wie *Donnerstagabend* oder *Dienstagmorgen* eingeführt. Diese wurden aus einem zufällig ausgewählten Wochentag und einer Tageszeit zusammengesetzt.

WORT Wochentag Tageszeit

Eine Vertiefung erfolgte anhand von abgeleiteten Substantiven.

drucke WORT Wortstamm Endung

Letztere Anweisung erzeugte am Bildschirm abgeleitete Substantive wie *Schicksal* oder *Reichtum*, aber auch „Gesundheit“ oder „Schicktum“. Den Kindern verdeutlichte dies Beispiel, dass die Angabe der syntaktischen Regel „ein abgeleitetes Substantiv setzt sich aus einem Wortstamm und einer Endung zusammen“ nicht ausreichend ist, sondern auch

semantische Vorschriften beachtet werden müssen.

Zur Vertiefung in einem anderen Kontext wurden in Englisch die Dialoge

- What's your name? My name is
- How old are you, ... ? I'm ... years old.
- Where are you from, ... ? I'm from

verschiedener Schülerpaare digital aufgezeichnet. In der zur Verfügung gestellten Dialog-Mikrowelt (vgl. Abbildung 4.8) mussten diese erst in Funktionen eingebaut werden, bevor sie über die Schaltflächen abgespielt werden konnten.

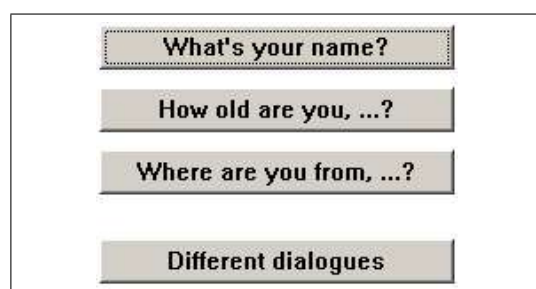


Abbildung 4.8: Benutzeroberfläche für englische Kurzdialege

Die Funktion `age` für den zweiten Dialog musste zum Beispiel den Namen der Audio-Datei des Dialogs zurückgeben.

```
PR age
  RG ZEL [ age_Ingrid age_Hans age_Dieter age_Natalie ]
ENDE
```

Sehr motivierend waren für die Schülerinnen und Schüler sowohl die Aufnahme der Dialoge als auch die Verwendung des Programms, da ein eigenes Produkt entstand und die Dialoge der anderen angehört werden konnten. Schwierigkeiten bei der Arbeit im Editor gab es nicht. Auch die Funktionen konnten nach einer Einführung im Klassenzimmer von fast allen Schülerpaaren selbstständig erstellt werden.

UML-Klassendiagramme

Zum Abschluss der Unterrichtseinheit *ebene Figuren* in Mathematik wurden noch einmal die verschiedenen Eigenschaften der Vierecke, Quadrate, Rechtecke, Parallelogramme und Rauten wiederholt, gemeinsame Eigenschaften herausgearbeitet und die Beziehungen zwischen den Figuren verdeutlicht. Zusammenfassend wurde dies in dem UML-Klassendiagramm in Abbildung 4.9 dargestellt.

Schon die Anzahl der Eigenschaften zeigte, dass die Spezialisierung der Figuren in entgegengesetzter Richtung zu den Pfeilspitzen zunahm. Die Schülerinnen und Schüler mussten selbst gleiche Eigenschaften in den Klassen farblich markieren und Beziehungen wie

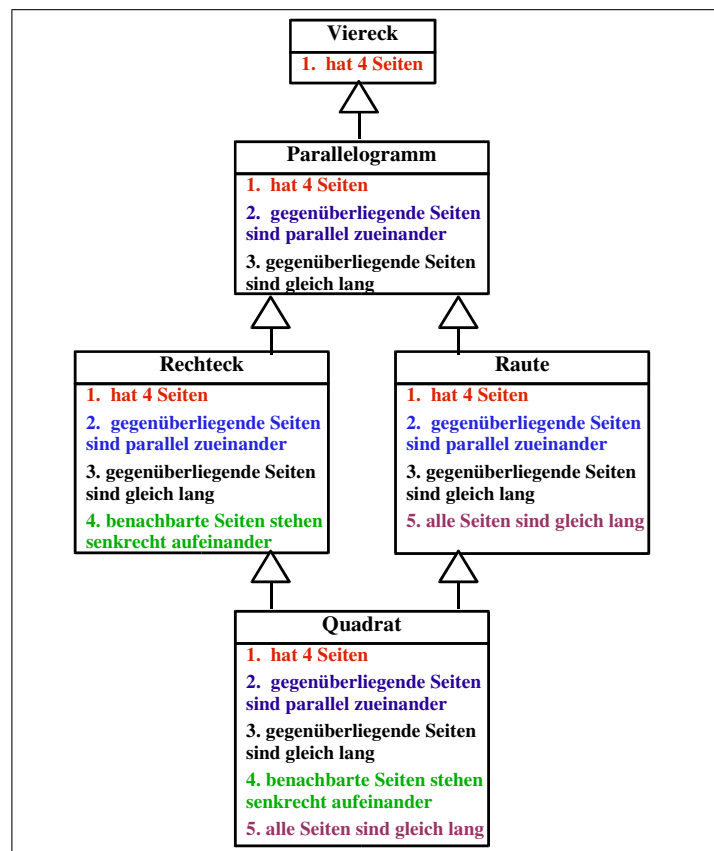


Abbildung 4.9: UML-Klassendiagramm – Ebene Figuren

„ein Rechteck ist auch ein Parallelogramm“ oder „ein Rechteck ist auch ein Viereck“ beschreiben. Die kompakte Darstellung bot einen sehr guten Überblick über die Zusammenhänge.

Modularisierung

Um die auditive Wahrnehmung der Schülerinnen und Schüler zu fördern, wurden im Musikunterricht mit Logo Melodien abgespielt. Die Lieder *Pfeifer Tim* und *Bunt sind schon die Wälder* lagen in vier bzw. sechs Module zerlegt vor, die mit den Befehlen `pt1`, ..., `pt4` und `bs1`, ..., `bs6` angehört werden konnten. Die meisten Melodiefragmente unterschieden sich im Rhythmus und der Tonbewegung deutlich. Den Charakter der Melodiebewegungen beschrieben die Kinder auf einem Arbeitsblatt. Anschließend komponierten sie aus den Musikmodulen eigene Melodien. Wie im Prozess der Modularisierung wurden die Kompositionen aus den Teillösungen zusammengesetzt. Aufgrund der sehr einfach zu bedienenden Melodie-Mikrowelt und der kompakt formulierten Arbeitsaufträge arbeiteten die Schülerinnen und Schüler in dieser Unterrichtseinheit sehr selbstständig.

Funktionsgeflecht

Die Konstruktion von Funktionsgeflechten bereitete den Schülerinnen und Schülern häufig große Schwierigkeiten, da sie nicht selbst in die Kommunikation involviert waren, sondern die Funktionen miteinander kommunizierten. In der Unterrichtseinheit zu den englischen „Is“-Fragen wurde mit einem Arbeitsblatt gearbeitet, das die Schülerinnen und Schüler mit Hilfe vieler detaillierter Fragen in folgenden vier Schritten an die Formulierung von Funktionsgeflechten heranführte:

1. Konkrete Ausprägung eines Aussagesatzes
 - This is Jenny
2. Aussagesätze mit Abstraktion der Namen
 - This is *Jenny*
 - This is *Ben*
 - This is *Trundle*
3. Fragen mit Abstraktion der Namen
 - Is *Jenny* a girl?
 - Is *Ben* a girl?
 - Is *Trundle* a girl?
4. Fragen mit Abstraktion der Namen und der Eigenschaften
 - Is *Jenny* a teacher?
 - Is *Ben* eleven?
 - Is *Trundle* a girl?

Auf den ersten beiden Stufen wurde der Datenfluss zwischen den Funktionen durch folgende zusammengesetzte Maschine beschrieben:

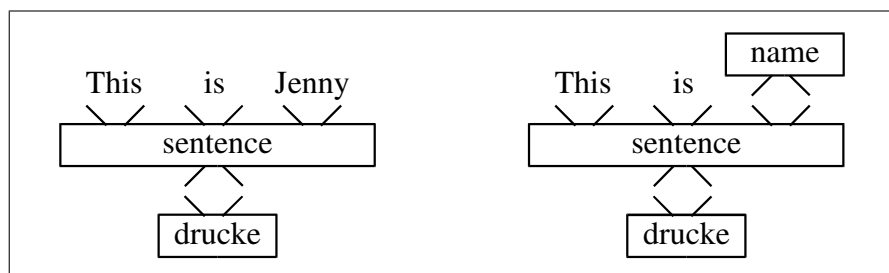


Abbildung 4.10: Zusammengesetzte Maschinen – Aussagesätze

Angeleitet durch das Arbeitsblatt wurden aus diesen die Funktion **name** und die Funktionsgeflechte

```
DRUCKE ( SENTENCE "This "is "Jenny )
```

und

```
DRUCKE ( SENTENCE "This "is name )
```

abgeleitet. Von Stufe zwei nach drei gab es keinerlei konzeptionelle Veränderungen. Aus diesem Grund mussten das Funktionsgeflecht in Logo und die Komposition der Maschinen für die Sätze der Stufe drei ohne Hilfestellung entwickelt werden.

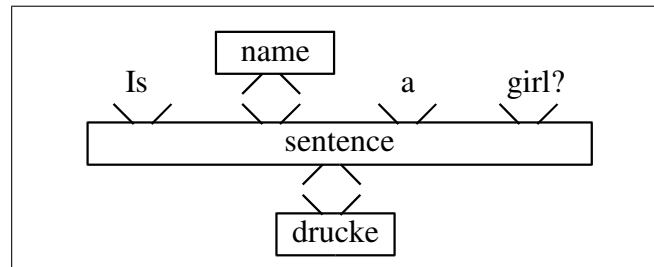


Abbildung 4.11: Zusammengesetzte Maschinen – Abstraktion der Namen in Fragen

Einige der von Logo automatisch generierten Sätze und Fragen wurden immer wieder auf das Arbeitsblatt eingetragen und dort beantwortet. Im letzten Schritt erfolgte eine Abstraktion der Namen und Eigenschaften. Zunächst notierten die Schülerinnen und Schüler die Eigenschaften auf Englisch, erstellten dann eine Funktion `characteristic` und testeten schließlich das Logo-Funktionsgeflecht.

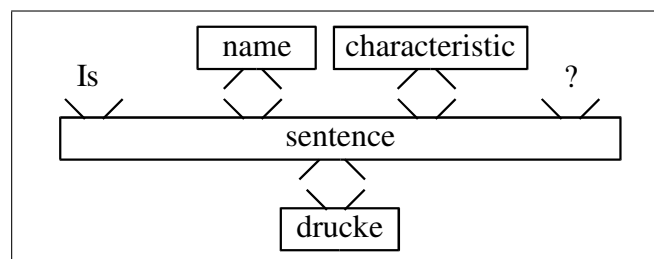


Abbildung 4.12: Zusammengesetzte Maschinen – Abstraktion der Namen und Eigenschaften in Fragen

In allen Schritten erarbeiteten die Kinder die Syntax der Sätze und Fragen sowohl als Funktionsgeflecht als auch als zusammengesetzte Maschine.

Die Unterrichtseinheit umfasste insgesamt fünf Arbeitsblätter, mit denen sich die Schülerinnen und Schüler selbstständig beschäftigten. Einige Kinder waren allerdings nicht bereit, die Aufgabenstellungen schrittweise zu lesen und zu bearbeiten. Sie riefen häufig die Forscherin und ließen sich die Aufgaben erklären. Rückblickend wäre es besser gewesen, für die verschiedenen Schritte eigene Arbeitsblätter zu erstellen und sie einzeln auszugeben. Der Umfang der Materialien würde dann nicht schon im Vorfeld der Bearbeitung zu einer negativen Arbeitshaltung führen.

Algorithmus

Bei der Umwandlung von römischen Zahlen ins Zehnersystem mussten die Schülerinnen und Schüler einen Algorithmus in schriftlicher Form und als Programmablaufplan lesen

und maschinell abarbeiten. Als Einführung wurden zunächst die römischen Zahlzeichen M, D, C, L, X, V und I vorgestellt und einige Zahlen aus dem Zehnersystem in das römische System umgewandelt. Sie waren so gewählt, dass die Zahlzeichen absteigend waren und somit dekodiert als Summand notiert werden konnten.

$$\text{LXVI} \quad 50 + 10 + 5 + 1 = 66$$

Anschließend wurde die Umwandlung exakt beschrieben.

Regel: Stehen die Zahlen in absteigender Größe von links nach rechts, so werden ihre Werte addiert.

Ausnahme: Stehen die Zeichen I, X oder C vor einem Zeichen mit höherem Wert, so wird ihr Wert nicht addiert, sondern subtrahiert.

Anhand von zwei Beispielen wurden die Regel und Ausnahme veranschaulicht und dann allein von den Schülerinnen und Schülern geübt. Einige Schwierigkeiten traten auf, da die Abfolge der Anwendung der Regel und Ausnahme nicht vorgegeben war. Dies motivierte jedoch die Beschreibung des folgenden Algorithmus:

1. Hat die römische Zahl nur 1 Zahlzeichen?
 - a) Falls deine Antwort *ja* ist, so schreibe den Wert des letzten römischen Zahlzeichens ganz links als Addition in deine Rechnung und führe die Berechnung durch. Dadurch erhältst du das Ergebnis.
 - b) Falls deine Antwort *nein* ist, arbeite bei 2. weiter.
2. Schaue dir in der römischen Zahl das Zahlzeichen ganz links an. Ist es kleiner als das Zahlzeichen direkt rechts daneben?
 - a) Falls deine Antwort *ja* ist, so schreibe den Wert des römischen Zahlzeichens ganz rechts als Subtraktion in deine Rechnung.
 - b) Falls deine Antwort *nein* ist, so schreibe den Wert des römischen Zahlzeichens ganz links als Addition in deine Rechnung.
3. Streiche das erste Zahlzeichen aus der römischen Zahl und fange wieder bei 1. an.

Längere Texte wurden häufig nicht oder nicht genau genug gelesen. Deshalb wurde der Algorithmus zusätzlich als Programmablaufplan (siehe Abbildung 4.13) angeboten.

Wie beim Algorithmus in Textform mussten die Schülerinnen und Schüler auch im Programmablaufplan den Pfeilen folgen, bei Ja-Nein-Entscheidungen die entsprechende Alternative wählen und Handlungen ausführen bis das Ende erreicht wurde. Zur Festigung wandelten sie in einer weiteren Übungsstunde noch einige vorgegebene und selbst konstruierte römische Zahlen in das Zehnersystem um. Die Ergebnisse wurden mit der in der Mikrowelt der römischen Zahlen vorhandenen Funktion `im_Zehnersystem` überprüft. Zusätzlich bekamen sie die Aufgabe, römische Zahlen zu finden, die von der Funktion `im_Zehnersystem` entweder nicht richtig umgerechnet wurden oder nicht zulässig waren. Im ersten Fall konnten sie keinen Fehler finden. Allerdings wurden Zahlen wie *VIC*

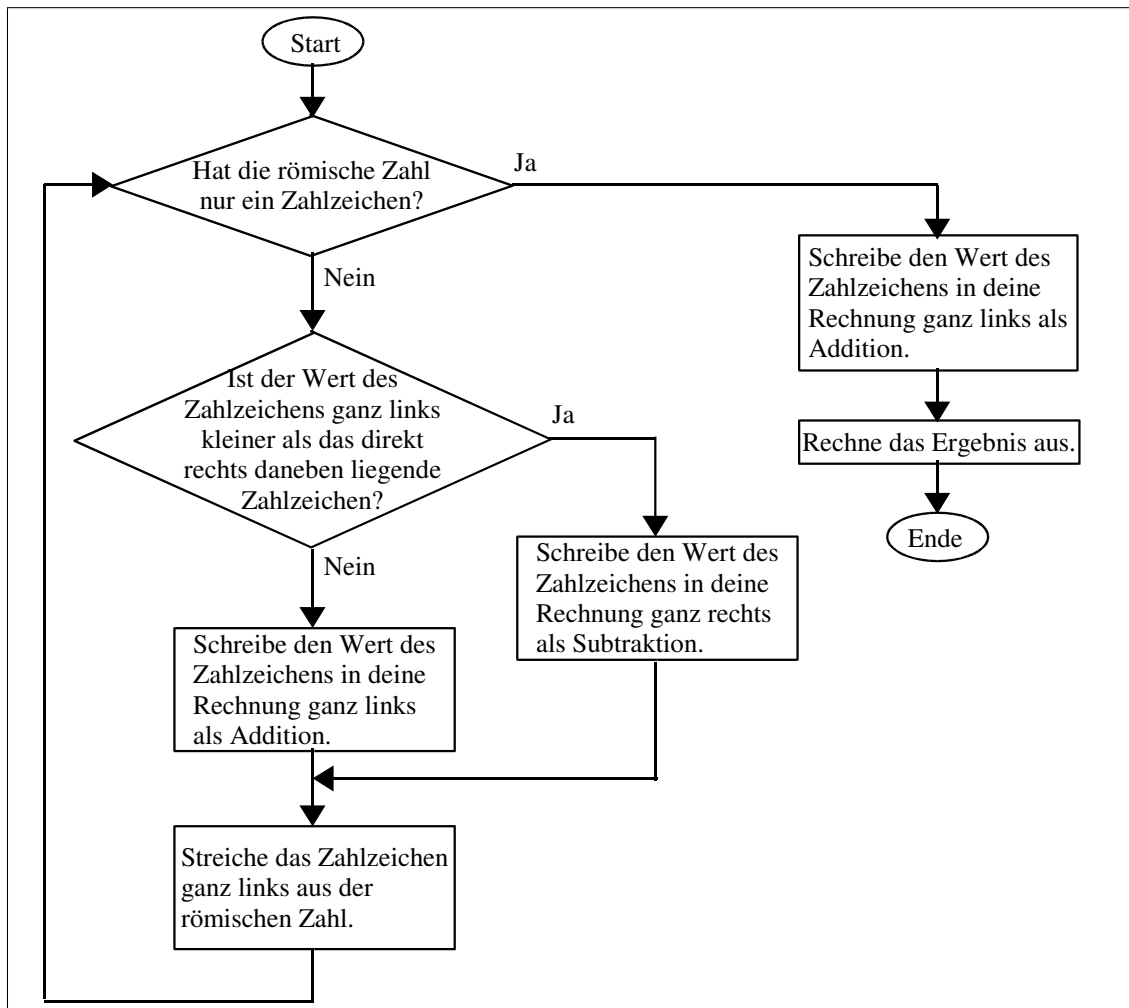


Abbildung 4.13: Programmablaufplan – Umwandlung römischer Zahlen

in 104 oder XXXXXXX in 70 konvertiert, obwohl es sich um keine korrekt gebildeten römischen Zahlen handelte. Wie schon bei der Verwendung des englischen Vokabeltrainers sollte auch hier eine Sensibilisierung für die Schwächen von Computerprogrammen geweckt werden.

Die Unterrichtseinheit war sehr erfolgreich, da sowohl mit Hilfe der textuellen Beschreibung des Algorithmus als auch mit dem Programmablaufplan viele Schülerinnen und Schüler die Aufgaben lösen konnten.

Äquivalenz von Repräsentationsformen

In Musik wurde ein weiteres Mal mit Musikbausteinen gearbeitet. Ausgewählt wurde die Melodie *Ein Vogelfänger* des Papageno aus der Oper *Die Zauberflöte* von Wolfgang Amadeus Mozart, da die Kinder die Oper in einer konzertanten Aufführung besuch-

ten. Der Melodieverlauf einzelner Teile musste sowohl textuell als auch mit Hilfe einer Tonhöhenlinie beschrieben werden. Dies bereitete die Schülerinnen und Schüler auf die Darstellung der Noten im Fünfliniensystem vor. In Abbildung 4.14 sind zwei der vier Musikbausteine dargestellt. Sie unterscheiden sich sowohl in der Länge als auch im Melodieverlauf.

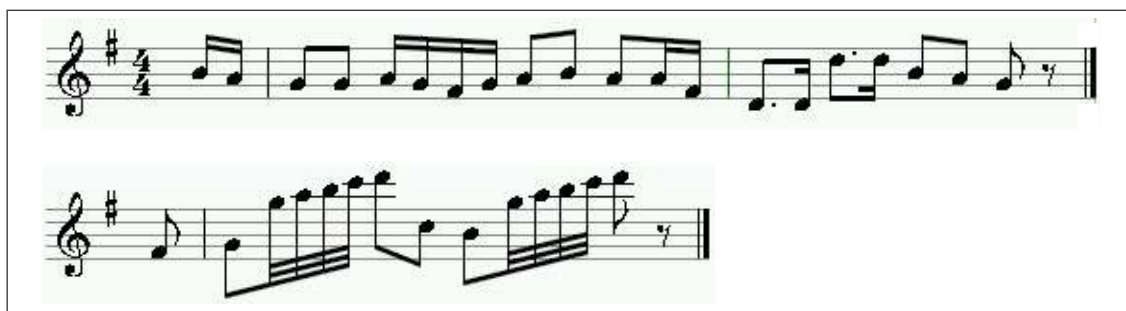


Abbildung 4.14: Melodieverlauf im Fünfliniensystem

Eine weitere Repräsentationsform waren Balkendiagramme. Die Tonhöhe wurde in der Höhe und die Tonlänge in der Breite der Balken kodiert. Die Musikmodule lagen also in eigenen textuellen Beschreibungen mit Angabe der Melodien in Form grafischer Linienverläufe, in Form von Audiodaten, im Fünfliniensystem und als Balkendiagramme vor. Die Melodiebausteine in den verschiedenen Repräsentationsformen mussten einander zugeordnet werden. Eine Vertiefung erfolgte an drei weiteren Modulen aus dem *Vogelfänger*.

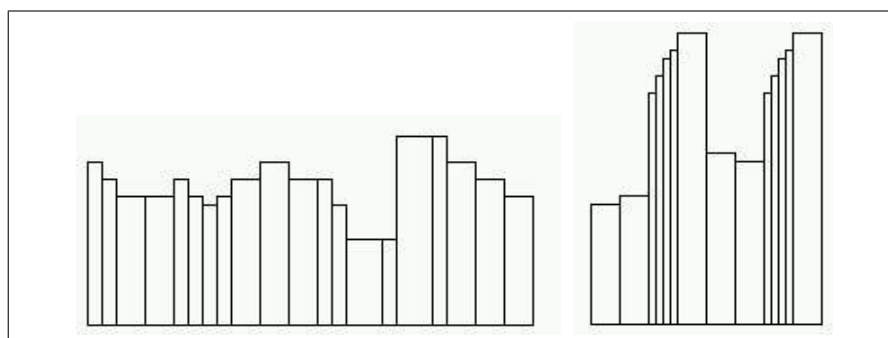


Abbildung 4.15: Melodieverlauf als Balkendiagramm

Für die meisten Kinder stellte die Aufgabe eine motivierende Herausforderung dar. Konzentriert setzten sie ihren Höreindruck mit der Notation in Beziehung und erkannten die Unterschiede in den Melodieverläufen.

Abstraktion

Zu den Inhalten der Unterrichtseinheit „Is“-Fragen in Englisch wurde ein Test geschrieben. Maschinen mussten in Befehle oder Funktionen aufgeteilt, Funktionen in Logo for-

muliert, zusammengesetzte Maschinen in Logo-Funktionsgeflechte übersetzt und englische Sätze, die eine zusammengesetzte Maschine erzeugte, aufgeschrieben werden. In der letzten Aufgabe wurden aufgrund der Analyse der Fragen

Is the desk green ?
Is the chair red ?
Is the pencil blue ?
Is the school bag yellow ?

die beiden Module *object* und *colour* identifiziert und als Funktionen implementiert. Die zusammengesetzte Maschine

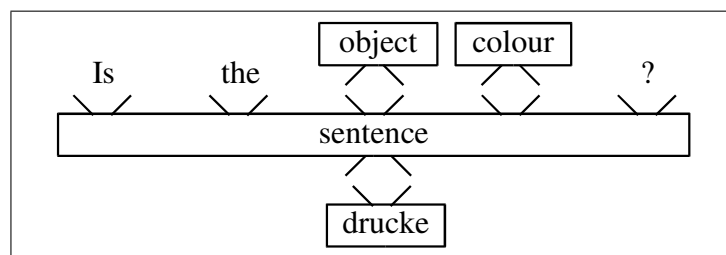


Abbildung 4.16: Zusammengesetzte Maschinen – Englische „Is“-Fragen

implementiert als Funktionsgeflecht in Logo

```
drucke ( SENTENCE "Is "the object colour "? )
```

erzeugte dann die Fragen. Bei der Besprechung des Tests wurden parallel zu den zusammengesetzten Maschinen zum ersten Mal Syntaxdiagramme zur Veranschaulichung der Satzstrukturen entwickelt.

Die Testergebnisse waren sehr schlecht. Nicht alle Schülerinnen und Schüler konnten die Abstraktion erkennen und diese dann auch formal mit zusammengesetzten Maschinen oder Logo-Funktionen beschreiben. Möglicherweise scheiterten einige schon daran, dass sie für die englischen Wörter *desk*, *chair*, *pencil* und *school bag* keinen Oberbegriff auf Englisch kannten. Bei einer Wiederholung müsste auf jeden Fall sicher gestellt werden, dass alle Kinder das zur Lösung der Aufgabe notwendige Vokabular kennen.

Programmablaufpläne erstellen

Die Bildung des Komparativs eignete sich für die Konstruktion von Programmablaufplänen, da schrittweise die verschiedenen Ausnahmen in das Schaubild aufgenommen werden konnten. Zunächst wurden die Steigerungsformen der Adjektive *hell*, *dürr*, *stark*, *gut*, *reich*, *leise*, *fleißig* und *schön* gebildet. Diese waren als Positiv, Komparativ oder Superlativ vorgegeben, so dass jeweils die fehlenden Formen gefunden werden mussten. Für die regelmäßigen Adjektive formulierten die Schülerinnen und Schüler selbstständig die Bildungsregel und die Logo-Funktion *Komparativ*.


```

PR Komparativ :Positiv
  RG WORT :Positiv "er
ENDE

```

Nach der Hinzunahme des Adjektivs *leise* wurde der folgende Programmablaufplan erstellt

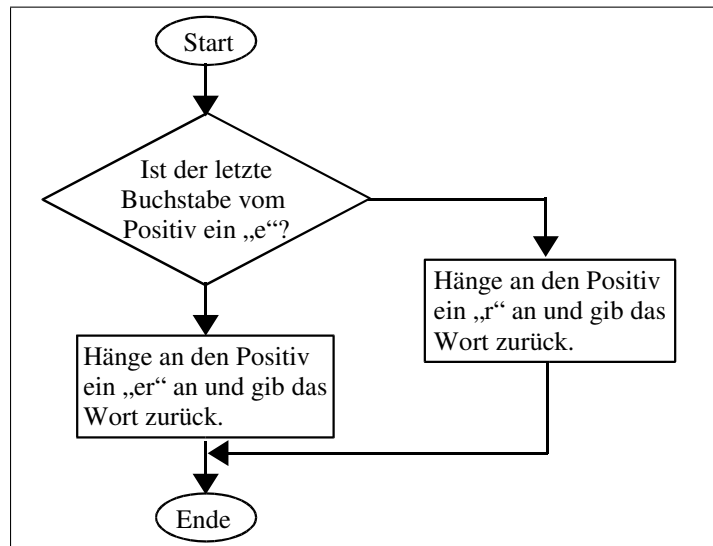


Abbildung 4.17: Programmablaufplan – Bildung des Komparativs

und die Funktion *Komparativ* um die Zeile

```

WENN GLEICH? LETZTER :Positiv "e [ RG WORT :Positiv "r ]

```

erweitert. Für die Hinzunahme der Adjektive *stark* und *gut* wurden die Bestandteile des Programmablaufplans auf einem Extrablatt vorgegeben. Diese mussten ausgeschnitten, in die richtige Reihenfolge gebracht und aufgeklebt werden. Gemeinsam wurden die Erweiterungen in die Funktion *Komparativ* eingebaut und die Zusammenhänge zwischen den Ja-Nein-Entscheidungen und Handlungen der Programmablaufpläne und dem **WENN**-Befehl mit seinen beiden Parametern **Vergleich** und **Handlung** wiederholt.

Das Verständnis des **WENN**-Befehls in Logo konnte im Laufe der Unterrichtseinheit verbessert werden. Im Computerraum hatten allerdings diejenigen Schülerinnen und Schüler, die die gemeinsam erarbeiteten Lösungen nicht korrekt von der Tafel abgeschrieben hatten, Schwierigkeiten, die syntaktisch komplexe Struktur des **WENN**-Befehls in Logo umzusetzen.

Automatisierung

In der Methodik-Woche Anfang Dezember war der Unterricht in den einzelnen Fächern aufgehoben. Da in dieser Zeit immer nur eine Hälfte der Klasse im Computerraum war,

konnte jedes Kind alleine an einem Rechner arbeiten. Zunächst hatten die Kinder die Aufgabe, mit Hilfe der Automatisierung Fadenbilder zu erstellen. In der Fadenbilder-Mikrowelt standen zwei Igel zur Verfügung, die sich gegenseitig ein Fadenknäuel zuwarfen und dadurch Linien auf dem Bildschirm hinterließen. Nur einer der beiden Igel war jeweils aktiv, hielt das Fadenknäuel in der Hand und konnte mit den Standardbefehlen **VORWÄRTS**, **RECHTS** etc. bewegt werden. Mit den Befehlen **erster_Igel** und **zweiter_Igel** wurde ein Igel aktiv gesetzt. Zum Werfen des Fadens gab es die beiden Befehle **Faden_zum_ersten** und **Faden_zum_zweiten**. Falls der entsprechende Igel nicht aktiv war, wurde eine Fehlermeldung ausgegeben. Bewusst wurde nicht nur ein Befehl **Faden_werfen** implementiert, der automatisch immer den Faden von dem gerade aktiven Igel warf. Dadurch wurden die Schülerinnen und Schüler wie in der Objektorientierung gezwungen, explizit Objekte auszuwählen.

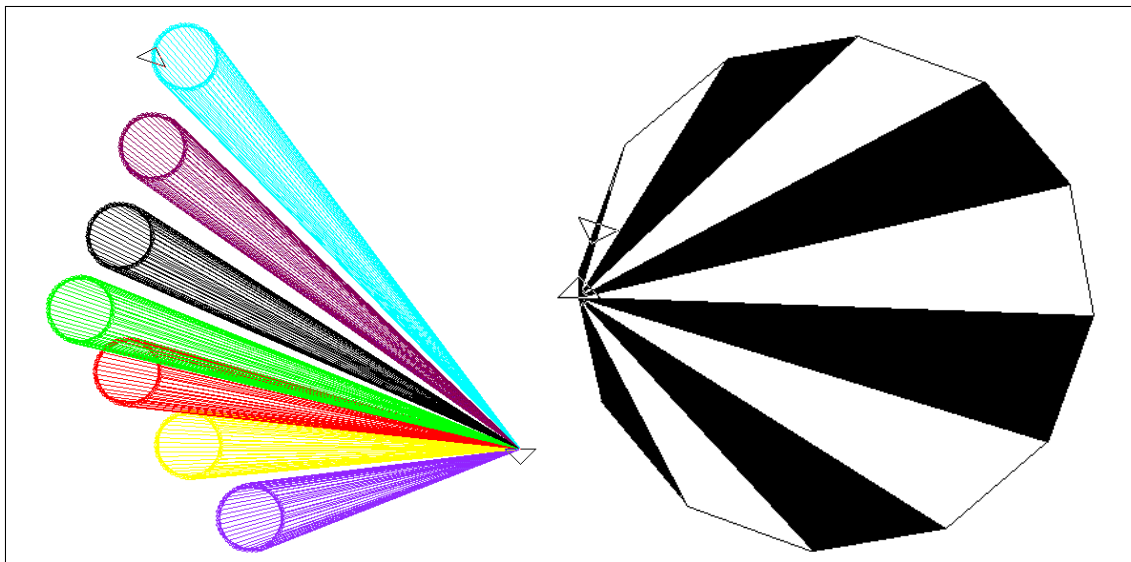


Abbildung 4.18: Fadenbilder

In der Arbeit der Schülerin links in Abbildung 4.18 ist zu sehen, dass der linke Igel beim Zeichnen der Figur, die das Kind Rohr nannte, auf einem Kreis gelaufen war und seinen Weg mehrmals unterbrochen hatte, um dem rechten Igel einen Faden zuzuwerfen. Ein Rohr wurde somit automatisch durch die Befehlsfolge

```
WIEDERHOLE 40 [ VORWÄRTS 33 RECHTS 53 Faden_zum_zweiten ]
```

erzeugt. Im Bild rechts ließ ein Schüler den Igel ein Zwölfeck ablaufen, an jedem Eckpunkt dem anderen Igel einen Faden zuwerfen und jede zweite Fläche schwarz einfärben. Auch ein Teil dieses Bildes wurde algorithmisch beschrieben.

Zur Gestaltung von Weihnachtskarten stellte die Weihnachtskarten-Mikrowelt einige Figuren in Form der Befehle **Quadrat**, **Rechteck**, **Dreieck**, **Kreis** und **Kerze** zur Verfügung.

Der Befehl `Igeltext` gab an der Position des Igels einen Text auf dem Bildschirm aus. Mit diesen sehr einfachen Mitteln der Weihnachtskarten-Mikrowelt wurden verschiedene Kompositionen entworfen (vgl. Abbildung 4.19).

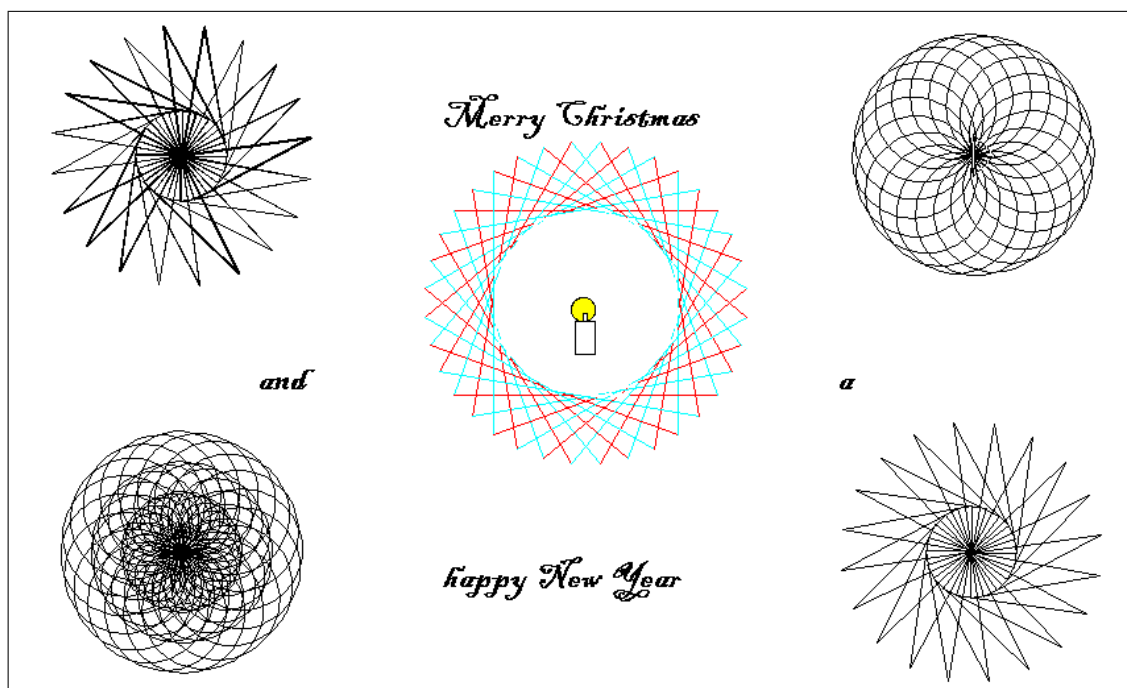


Abbildung 4.19: Weihnachtskarte

Sowohl bei den Fadenbildern als auch bei den Weihnachtskarten waren die Schülerinnen und Schüler motiviert und sehr produktiv. Es entstanden ganz unterschiedliche und teilweise sehr kreative Ergebnisse. Nicht alle automatisierten die Erstellung ihrer Bilder. Der kognitive Mehraufwand rechtfertigte für diese Kinder nicht die Zeitersparnis.

Fehleranfälligkeit von Computersystemen

Am Beispiel der Währungsumrechnung stellte sich für die Schülerinnen und Schüler sehr anschaulich dar, dass auch Ergebnisse, die der Computer berechnet, überprüft werden müssen. Zunächst waren die Preise eines Skipasses, einer Bratwurst, einer Cola und einer Rakete für ein Silvesterfeuerwerk in Schweizer Franken vorgegeben. Ohne Angabe eines Wechselkurses sollten die Preise in Euro geschätzt werden. Dabei mussten die Verhältnisse der Preise in Schweizer Franken und Euro in Beziehung gesetzt werden. Dies gelang nur einem Schüler. Die Berechnung der exakten Preise wurde gemeinsam mit Hilfe des Zweisatzes entwickelt.

Anhand des Szenarios, dass ein Mitarbeiter einer Bank beim Rücktausch von 360 Schweizer Franken in Euro statt Schweizer Franken amerikanische Dollar in den Computer eingegeben hat, musste ausgerechnet werden, ob die Bank mit dem Geschäft einen Gewinn

oder Verlust machte.

```
360 SF    sind    ... Euro
360 USD   sind    ... Euro
```

Der Umrechnungskurs der amerikanischen Dollar war mit 60 zu 47 angegeben. Anhand der Aufgabe wurde im Klassenverband analysiert, welche Funktionalität ein Computerprogramm zur Umrechnung von Währungen besitzen muss, an welchen Stellen der Umrechnung Fehler auftreten können und von wem die Fehler verursacht werden.

Sehr konzentriert und motiviert rechneten die Schülerinnen und Schüler die Währungen um. Der Kontext vermittelte ihnen das Gefühl, ein Rätsel lösen zu dürfen. Die Kreativität bei der Reflexion möglicher Fehlerquellen war sehr unterschiedlich.

Syntaxdiagramme

Im Themenkomplex *Satzlehre* in Deutsch wurden zahlreiche Satzbaupläne mit Hilfe von Syntaxdiagrammen formal beschrieben. Dabei konnten die Permutationsmöglichkeiten der einzelnen Satzglieder in der deutschen Sprache im Gegensatz zum Englischen sehr anschaulich dargestellt werden.

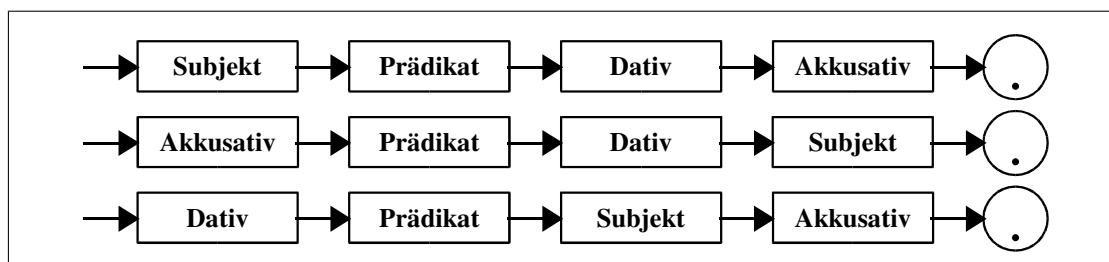


Abbildung 4.20: Stellung der Satzglieder in deutschen Sätzen

Aus den Syntaxdiagrammen leiteten die Schülerinnen und Schüler dann sehr leicht Logofunktionsgeflechte ab. Im Beispiel in Abbildung 4.21

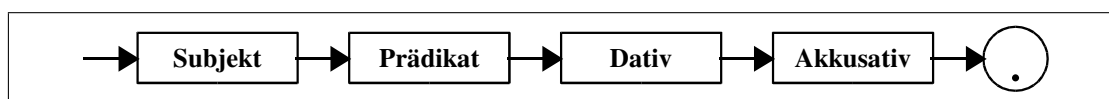


Abbildung 4.21: Syntaxdiagramm – Aussagesätze mit Dativ- und Akkusativobjekten

wurden die Rechtecke als Funktionen implementiert und das Gesamtdiagramm mit der Funktion SATZ

```
drucke ( SATZ Subjekt Prädikat Dativ Akkusativ ". . )
```

zusammengesetzt. Natürlich entstanden auch semantisch falsche Sätze, so dass über die Bedeutung von Sprache diskutiert wurde.

Einige Schülerinnen und Schüler konnten die Satzglieder zwar farbig richtig unterstreichen, eine Abstraktion in Form eines Syntaxdiagramms gelang ihnen jedoch nicht. Problematisch war im Rückblick, dass die Beispielsätze und die zu zeichnenden Syntaxdiagramme auf verschiedenen Seiten des Arbeitsblatts abgebildet waren.

Automatisierte Tests

Bis zu diesem Zeitpunkt hatten die Schülerinnen und Schüler verschiedene Programme entworfen, mit denen einzelne Zufallssätze oder auch Zufallswörter wie die abgeleiteten Substantive oder der Komparativ generiert wurden. Auch für unbestimmte Artikel englischer Substantive formulierten die Kinder Regeln, zeichneten Programmablaufpläne und programmierten Logo-Funktionsgeflechte. Die erste Version der Funktion `indefinite_article`, die den Artikel eines Substantivs zurückgab, wurde mit einigen Substantiven getestet.

```
drucke indefinite_article "answer
drucke indefinite_article "question
drucke indefinite_article [exercise book]
```

Die Funktion wurde dahingehend erweitert, dass zusammen mit dem unbestimmten Artikel auch das Substantiv zurückgegeben wurde. Dies hatte zur Folge, dass alle Testfälle noch einmal ausprobiert werden mussten. Um Zeit zu sparen und systematischer vorzugehen, erstellten die Schülerinnen und Schüler eine Funktion `Substantiv`, die die zu testenden Substantive enthielt und zufällig eines zurückgab. Mit dieser und dem WIEDERHOLE-Befehl konnte der Test der Funktion `indefinite_article` automatisiert werden.

```
WIEDERHOLE 15 [ drucke indefinite_article Substantiv neue_Zeile ]
```

Die Anzahl der Wiederholungen war so groß zu wählen, dass alle Substantive der Liste mindestens einmal ausgewählt wurden. Um besser die Übersicht zu behalten, wurde nach jeder Ausgabe eines Substantivs eine neue Zeile erzeugt.

Das sehr kleinschrittige Vorgehen unterstützte die Schülerinnen und Schüler bei der Erstellung ihrer Funktionen und den Tests. Trotzdem hatten einige bereits syntaktische Regeln von Logo wie das Schreiben des komplexen WENN-Befehls vergessen, so dass häufig Hilfestellungen gegeben werden mussten.

Objekte in unterschiedlichen Repräsentationsformen und ihre Umwandlung

Die Schülerinnen und Schüler erarbeiteten sich das Fünfliniensystem in Musik selbst in einem Test, der bewertet wurde. Die Begriffe *Fünfliniensystem*, *Linie*, *Notenschlüssel*, *Violinschlüssel*, *Taktstrich* und *Takt* waren in einem Text beschrieben und wurden gemeinsam gelesen. Danach mussten die Kinder für jeden Begriff ein Beispiel aufschreiben. In der zweiten Aufgabe ging es um die Unterscheidung von Attributwerten. Es musste die Anzahl der unterschiedlichen Noten, die sich aus den verschiedenen Tonhöhen und

Tonlängen ergab, in der Melodie in Abbildung 4.22 bestimmt werden.

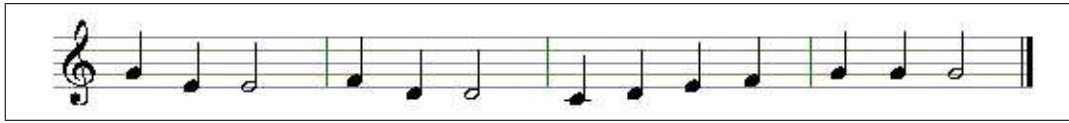


Abbildung 4.22: Melodie im Fünfliniensystem

Zusätzlich zu der bis dahin verwendeten Repräsentation der Töne als Zahlen lernten die Schülerinnen und Schüler die Buchstaben-Zahlen-Kombinationen c^1 , d^1 etc. und die Positionen der Noten im Fünfliniensystem kennen. In der dritten Aufgabe wurde eine Funktion `Buchstabe_aus_Zahl` zur Umwandlung der Zahldarstellung in Notennamen und ein Befehl `drucke_Note` zur Visualisierung eines Notennamens im Fünfliniensystem zur Verfügung gestellt. Diese wurden sowohl in Textform als auch mit den in Abbildung 4.23 dargestellten Maschinen präsentiert.

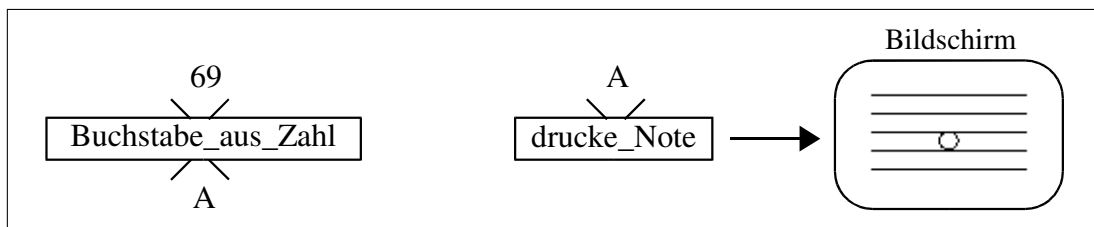


Abbildung 4.23: Maschinen zur Umwandlung von Tonrepräsentationen

Die Schülerinnen und Schüler mussten beide verwenden, um die Töne der C-Dur-Tonleiter von c^1 bis c^3 in einer Tabelle in allen Repräsentationsformen einzutragen. In drei weiteren Aufgaben wurde die Umwandlung zwischen den einzelnen Repräsentationsformen geübt. Sehr konzentriert arbeiteten die Kinder an den Arbeitsblättern. Niemand hatte Schwierigkeiten, Logo bzw. die angegebene Funktion und den Befehl als Werkzeug zu benutzen. Die Aufgaben lösten fast alle. Ein Grund dafür könnte die Bewertung des Tests sein, dessen Ergebnis in die Musiknote einging.

Grammatik

In Englisch wurden Aussagesätze der Form

Jenny has got a silly tortoise.

The twins have got a great dog.

in Sätze mit dem s-Genitiv

Jenny's tortoise is silly.

The twins' dog is great.

umgeformt. Die syntaktische Regel der Grammatik für die Aussagesätze mit „has got“ war durch folgendes Syntaxdiagramm vorgegeben:

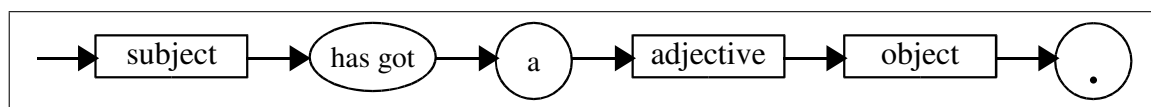


Abbildung 4.24: Syntaxdiagramm – Has-Sätze

Die einzelnen Bestandteile mussten in den Sätzen farbig unterstrichen werden. Für die Sätze mit „have got“ zeichnen die Kinder selbst ein Syntaxdiagramm. Außerdem bildeten sie neue has/have-Sätze und wandelten diese in Genitiv-Sätze um. Zum Abschluss entwarfen sie noch für alle Genitiv-Sätze ein Syntaxdiagramm. So wurde die Grammatik nicht nur aus Syntaxdiagrammen gelernt, sondern die Schülerinnen und Schüler konstruierten die Grammatik selbstständig aus den Beispielsätzen durch Abstraktion und formalisierten sie mit Hilfe des informatischen Beschreibungsmittels.

Die Schülerinnen und Schüler arbeiteten sehr gut mit. Aus diesem Grund war es erstaunlich, dass, wie in Kapitel 6.4.5 weiter ausgeführt wird, die Bearbeitung der Aufgabenblätter sehr unvollständig und teilweise trotz Besprechung nicht korrekt war.

Erweiterung von Repräsentationsformen bei Einführung neuer Attribute

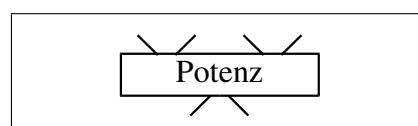
Wurden Melodien mit dem Befehl `spiele` abgespielt, so mussten die einzelnen Töne als Zahlen angegeben werden. Diese wie auch die Notennamen c^1 , d^1 etc. kodierten keine Tonlängen. Da nur Achtel, Viertel, halbe- und ganze Noten stoffrelevant waren, wurden zur Erweiterung der Repräsentationsform einfach die Anfangsbuchstaben a , v , h und g vor die Bezeichnungen gestellt. Die veränderte Datenstruktur erforderte zum Abspielen der Töne den neuen Befehl `spiele_Noten`. Das Beispiel

```
spiele_Noten [ vg1 ve1 he1 vf1 vd1 hd1 vc1 vd1 ve1 vf1 vg1 vg1 hg1 ]
```

kodiert die ersten vier Takte der Melodie *Hänschen klein*. Die Schülerinnen und Schüler erkannten, dass Prozeduren mit ganz speziellen Repräsentationsformen von Daten arbeiten, die nicht beliebig ausgetauscht werden können.

Attribute

Wie die Töne wurden auch die Potenzen als Klasse mit den Attributen *Exponent* und *Basis* eingeführt. In einer Menge von Potenzen kennzeichneten die Schülerinnen und Schüler die Quadratzahlen, Kubikzahlen und Zehnerpotenzen in unterschiedlichen Farben. Anschließend wurde ihnen eine Logo-Funktion **Potenz**, mit der beliebige Potenzen berechnet werden konnten, in Maschinendarstellung präsentiert.

Abbildung 4.25: Funktion **Potenz** als Maschine

Aus der Darstellungsform ging nicht hervor, in welcher Reihenfolge die Basis und der Exponent anzugeben waren. Aus diesem Grund wurde die Funktion mit dem Syntaxdiagramm

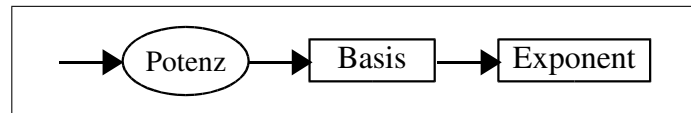


Abbildung 4.26: Funktion **Potenz** als Syntaxdiagramm

weiter präzisiert. Zum ersten Mal wurden Syntaxdiagramme nicht zur Beschreibung von Sprache, sondern von mathematischen Strukturen verwendet. Dies verdeutlichte die Universalität des Beschreibungsmittels. Im Gegensatz dazu sahen die Schülerinnen und Schüler, dass ihre eigenen Funktionen **Quadratzahl**, **Kubikzahl** und **Zehnerpotenz** wegen der Fixierung eines Attributs auf einen speziellen Wert nur einen Eingabeparameter hatten. Da die Kinder zu diesem Zeitpunkt sowohl Maschinen als auch Syntaxdiagramme als Beschreibungsmittel formaler Strukturen kannten, konnten sie die Funktion **Potenz** selbstständig zur Überprüfung der berechneten Ergebnisse verwenden.

Codierung

Das Konzept der Codierung wurde spielerisch am Beispiel der Entschlüsselung deutscher Wörter eingeführt. Die entsprechenden Dekodierungsalgorithmen waren als Programmablaufpläne formuliert, die bereits eine relativ komplexe Struktur besaßen. Durch sehr konzentriertes Abarbeiten des Ablaufs und der Befolgung der Anweisungen konnten die Schülerinnen und Schüler die Wörter dekodieren. Um das Codierungsprinzip zu erfassen, mussten in einem zweiten Schritt selbst Wörter kodiert werden. Dies erforderte, die Programmablaufpläne gedanklich rückwärts zu lesen. Dies schafften nur sehr wenige.

Äquivalenz formaler Beschreibungsformen

Bei der automatischen Generierung englischer Sätze in der *present progressive*-Form standen zusammengesetzte Maschinen, Logo-Funktionsgeflechte und Syntaxdiagramme im Vordergrund. Zunächst schrieben die Schülerinnen und Schüler Sätze der Form

My mother is reading a book.
 My father is cleaning the dishes.
 My brother is helping in the kitchen.

auf. Daraufhin abstrahierten sie die Sätze in Form des Syntaxdiagramms,

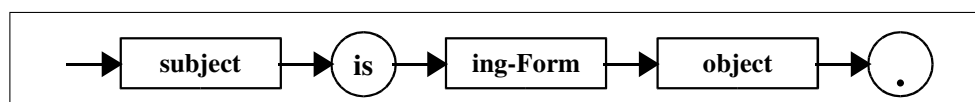


Abbildung 4.27: Syntaxdiagramm – Sätze im *present progressive*

der zusammengesetzten Maschine

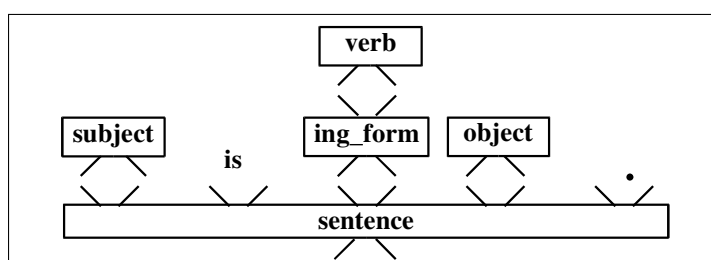


Abbildung 4.28: Zusammengesetzte Maschinen – Sätze im *present progressive*

und dem Logo-Funktionsgeflecht.

```
( sentence subject "is ing_form verb object ". )
```

Mit allen drei Beschreibungsformen konnte die Struktur der englischen Sätze im *present progressive* äquivalent dargestellt werden. Die erste Version der Funktion `ing_form` der Kinder gab für regelmäßige Verben die entsprechende ing-Form zurück. Da die Funktion `verb` nur die Verben *read*, *clean*, *help*, *draw*, *listen* und *work* enthielt, lieferte das Logo-Funktionsgeflecht korrekte Sätze.

Nach der Erweiterung der Liste um die Verben *write*, *make* und *come* stellte sich beim Testen heraus, dass die ing-Formen der neuen Verben falsch waren. Diese Sonderfälle beschrieben die Schülerinnen und Schüler algorithmisch als Text, integrierten sie in den Programmablaufplan der Gesamtlösung und bauten sie anschließend in die Logo-Funktion `ing_Form` ein. Dabei erkannten sie, dass es sich auch bei diesen Beschreibungen um äquivalente Repräsentationsformen handelte, die sich für verschiedene Problemstellungen unterschiedlich gut eigneten. Im Programmablaufplan konnte die Unterscheidung zwischen dem allgemeinen Fall und Sonderfall sehr deutlich und kompakt dargestellt werden.

Hierarchische Darstellung mit Mindmaps

Bei der Wiederholung der Potenzen in Mathematik wurden die Inhalte, Fragestellungen und konkreten Aufgaben in der in Abbildung 4.29 dargestellten Mindmap den Schülerinnen und Schülern präsentiert. Im Zweig der *besonderen Potenzen* und *Aufgabentypen* konnten an die Blätter der Hierarchie Beispiele geschrieben werden. Der geschlossene Ast der Aufgaben enthielt Arbeitsanweisungen zur Darstellung von Dezimalzahlen als Potenzen.

Dies Beispiel zeigte den Schülerinnen und Schülern, wie Lernstoffe übersichtlich strukturiert, wichtige Inhalte hervorgehoben und Aufgabenstellungen selbst entwickelt werden können. Mindmaps wurden auch in anderen Fächern eingesetzt, so dass verschiedene Einsatzformen deutlich wurden.

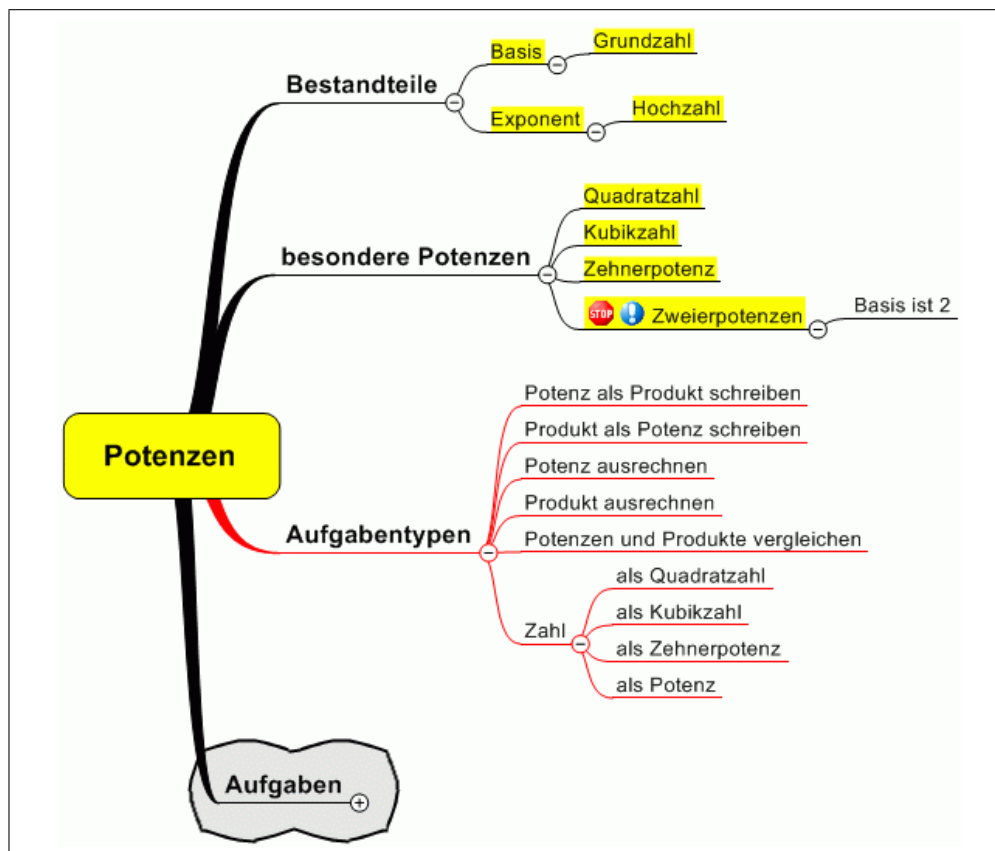


Abbildung 4.29: Mindmap zu Potenzen

Objekte und Ereignisse an der Programmoberfläche

Die Einführung in die Objektorientierung mit der Spezifizierung von Objekten und Ereignissen erfolgte bei der Einführung von Imagine Logo. Als Anwendung wurde ein Spiel zum Erlernen der Zehnerpotenzen eingesetzt, das von den Schülerinnen und Schülern angepasst werden musste (vgl. Abbildung 4.30).

Durch Anklicken einer fliegenden Zahl wurde für eine vorgegebene Zahl der Exponent der Zehnerpotenz ausgewählt. Bei der ersten Demonstration des Programms durch die Forscherin identifizierten die Schülerinnen und Schüler im Klassengespräch die verschiedenen Objekte. Sie unterschieden die Textfelder, die fliegenden Zahlen und die Schaltfläche. Auf einem Arbeitsblatt trugen sie diese zusammen mit den englischen Bezeichnungen in eine Tabelle ein. In einem zweiten Schritt beschrieben sie die Möglichkeit der Interaktion. Sie hielten die unterschiedlichen Reaktionen des Programms beim Drücken der Schaltfläche *Neue Aufgabe* und dem Anklicken der richtigen und falschen Exponenten schriftlich fest.



Abbildung 4.30: Lernspiel Zehnerpotenzen

Die Mikrowelt der Zehnerpotenzen, mit der die Schülerinnen und Schüler anschließend eigenständig arbeiteten, enthielt alle Objekte an der Oberfläche. Zum Erlernen des Unterschieds zwischen den Ereignissen `onPush` und `onClick` der Klassen `Schaltfläche` und `Igel` musste die komplette Interaktion selbst programmiert werden. Die Methoden zur Änderung des Schaltflächen-Objekts und der Zahlen-Objekte wurden mit der Maus ausgewählt. Mit Hilfe der Befehle

```
neue_Zehnerpotenz    starte_fliegende_Zahlen
```

und

```
prüfe_Ergebnis_Zehnerpotenz
```

steuerten die Schülerinnen und Schüler den Spielverlauf in den Ereignissen `onPush` und `onClick`. Alle drei Befehle waren bereits in der Mikrowelt enthalten.

Hierarchie

Die Struktur einer Hierarchie, dargestellt in einem Baum mit Wurzel, mehreren Ebenen und Blättern, wurde in Deutsch in eine Rechtschreibübung eingebettet. Die Schülerinnen und Schüler ordneten die Verben und Substantive in die Hierarchie in Abbildung 4.31 ein, erfanden zu den Bildern in Abbildung 4.32 eine Geschichte, in der mindestens zehn der Begriffe vorkamen, und schrieben sie auf die drei vorgegebenen Bilderseiten in Imagine Logo. Dabei lernten sie die Attribute *Schriftart*, *Schriftgröße*, *Schriftschnitt* und *Schriftfarbe* der Klasse `Text` (engl. `text box`) kennen und experimentierten mit deren Attributwerten.

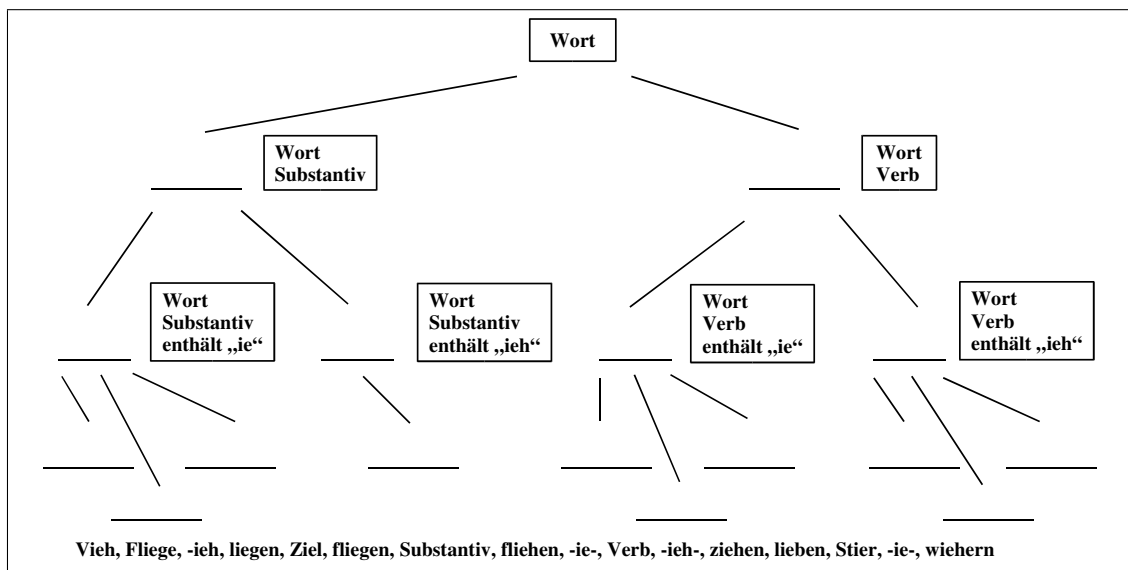


Abbildung 4.31: Begriffshierarchie

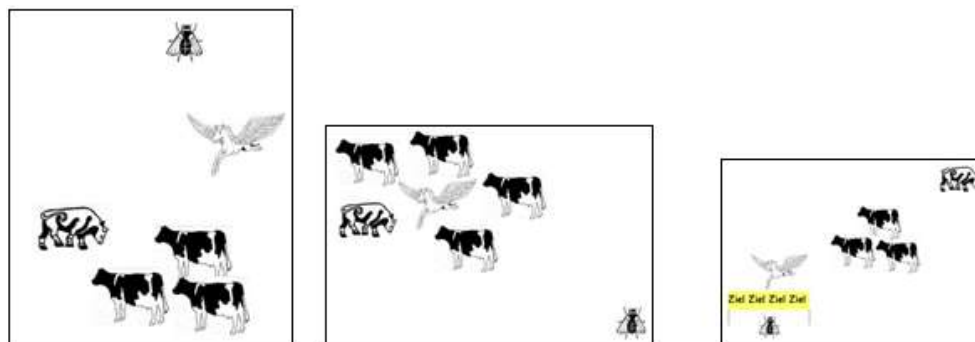


Abbildung 4.32: Bildergeschichte

Insgesamt zeigte sich, dass die Kinder ohne Einführung mit dem in Imagine Logo eingebetteten sehr einfachen Textsystem arbeiten konnten. Es war nicht notwendig, neues Anwendungswissen aufzubauen, da es bereits durch die Arbeit an den informatischen Grundkonzepten und mit ihrer Anwendung in den Logo-Systemen erworben worden war.

Objektorientierte Programmierung

Die Schülerinnen und Schüler vertieften die Syntax englischer Sätze in der Verlaufsform anhand einer Bildergeschichte. Bei der Animation einer Figur der Geschichte wurde in die objektorientierte Programmierung eingeführt. In dem in Abbildung 4.33 dargestellten Bild (Hauß & al., 1997) der Geschichte läuft ein Junge gerade von der Schule nach Hause.

Diese Situation wurde im Klassenraum mit einer Schülerin simuliert. Ihre Mitschülerinnen und Mitschüler hatten die Aufgabe, sie vom Lehrerpult zur Tür und wieder zurück



Abbildung 4.33: Animation einer Figur

zu kommandieren. Sie durfte sich nur bewegen, wenn sie mit ihrem Namen angesprochen wurde. Außerdem musste genau gesagt werden, was sie tun sollte. Der Junge in der Bildergeschichte erhielt den Namen *Jim*. Mit dem Befehl

```
Jim'FORWARD 20
```

konnte auch er vorwärts bewegt werden. Da Jim ständig weiterlaufen sollte, musste ihm der Befehl immer wieder neu erteilt werden.

```
FOREVER [ Jim'FORWARD 20 ]
```

Die Schülerinnen und Schüler stellten fest, dass er durch diese Anweisung viel zu schnell rannte. Sie mussten also zwischen jedem *Schritt* eine kurze Pause einbauen.

```
FOREVER [ Jim'FORWARD 20 WAIT 100 ]
```

Die Bewegung von Figuren faszinierte die Kinder auch in vielen weiteren Unterrichtseinheiten. Da die programmiersprachliche Umsetzung in Logo sehr intuitiv und einfach zu formulieren war, konnten alle Kinder die Animationen umsetzen.

Konstruktion eines Netzwerks

Aus einer textuellen Beschreibung von 28 verwandtschaftlichen Beziehungen, die in Englisch im Genitiv formuliert waren, mussten die Schülerinnen und Schüler in dieser Unterrichtseinheit einen Familienstammbaum konstruieren.

```
Waltraud's mother is Hulda.
Hulda's husband is Gustav.
Waltraud's sister is Charlotte.
...
```

Anders als in einer Hierarchie hatten die Knoten hier fast immer zwei Vorgänger, den Vater und die Mutter. Es gab keinen ausgezeichneten Knoten, so dass die Struktur des Stammbaums ein Netzwerk war. Der Begriff des Netzwerks wurde nicht eingeführt. Da es im Gegensatz zur Hierarchie keine natürliche Anordnung der Knoten gibt, war es

besonders wichtig, dass die Schülerinnen und Schüler in ihren Darstellungen selbst eine klare Struktur aufbauten und die verwandtschaftlichen Beziehungen durch Verbindungen verdeutlichten. Dies gelang, wie in Kapitel 6.4.4 beschrieben wird, noch nicht allen.

4.2 Projektarbeit

Nachdem die einzelnen informatischen Grundkonzepte im Kontext der verschiedenen Fachinhalte eingeführt waren, hatten die Schülerinnen und Schüler während der Projektarbeit an der Rittergeschichte die Möglichkeit,

- an einem großen Projekt zu arbeiten, in dem sie sich ihre Zeit selbst einteilten,
- in einer Arbeit die Inhalte von vier verschiedenen Fächern zu erlernen,
- die informatischen Grundkonzepte anzuwenden und zu vertiefen,
- in kreativer Arbeit ein ästhetisch ansprechendes Produkt zu erstellen und
- eine Präsentation vorzubereiten (Wursthorn, 2005b).

Um sie nicht zu überfordern, erhielten sie zu Beginn der Projektarbeit ein Arbeitsblatt, auf dem die Pflichtteile des Projekts in Form von Handlungen abgebildet waren. Diese sollten sie nutzen, um zur Planung der Reihenfolge der verschiedenen Seiten ihrer multimedialen Geschichte einen Programmablaufplan zu erstellen. Außerdem bekamen sie nach etwa zwei Wochen eine Checkliste, auf der sie die Anforderungen einzeln abhaken konnten.

In den ersten beiden Wochen wurden in Doppelstunden die Fachthemen achsensymmetrische Figuren (Mathematik), Genitiv (Englisch), Gedicht (Deutsch) und Rhythmus (Musik) in kurzen Einführungen aufgegriffen und das für die Umsetzung in Logo notwendige Wissen instruktional vermittelt. Im Anschluss daran arbeiteten die Schülerinnen und Schüler völlig selbstständig an ihren Produkten. Außer der Rittergeschichte, die als Hausaufgabe geschrieben wurde, und den Bildern, die im Kunstunterricht gezeichnet und von der Forscherin eingescannt wurden, entstanden die Projektergebnisse ausschließlich im Unterricht. Sie wurden nach fünf Wochen den Eltern präsentiert.

Somit genügte das Thema der Rittergeschichte und die Durchführung der Arbeit fast allen von Gudjons (1997) beschriebenen Merkmalen eines Projekts. Der Situationsbezug war nicht nur auf das Schreiben einer Geschichte beschränkt, sondern durch ihre multimediale Gestaltung konnten auch Aspekte aus anderen Fächern, wie zum Beispiel die achsensymmetrische Ritterburg, integriert werden. Das Thema der Rittergeschichte selbst orientierte sich sowohl an den kämpferischen Interessen der Jungen als auch den gestalterischen der Mädchen. Da es sich bei der Rittergeschichte um das erste Projekt der Schülerinnen und Schüler handelte, mussten sie bei der zielgerichteten Projektplanung und der Selbstorganisation mit Hilfe der oben beschriebenen Checkliste unterstützt und gelenkt werden. Nach einer gewissen Zeit arbeiteten sie jedoch in völliger Selbst-

verantwortung. Die Einbeziehung vieler Sinne war durch die Arbeit am Computer auf das Sehen und Hören eingeschränkt. Soziales Lernen übten die Schülerinnen und Schüler durch die Zusammenarbeit in Paaren. Besonders bei den kreativen Tätigkeiten mussten sie aufgrund unterschiedlicher Vorstellungen häufig Kompromisse aushandeln. Die Interdisziplinarität und Produktorientierung des Projekts war durch die Integration der Fächer Mathematik, Englisch, Deutsch und Musik und die Erstellung einer Präsentation vorgegeben.

Neue informatische Grundkonzepte		
Fachinhalt	Mikrowelt	Wiederholung
Modularisierung		
Mathematik: Achsensymmetrische Burg	Spiegel-Mikrowelt	Befehle, Befehlsfolgen, Planung <i>komplexer</i> Igelbewegungen
Netzwerk		
Englisch: Stammbaum der Ritterfamilie		Klasse <i>Textfeld</i> , <i>Igel</i>
Objektorientierung		
Englisch: Rätsel zum Stammbaum der Ritterfamilie		Klasse <i>Textfeld</i> , <i>Schaltfläche</i> , objektorientierte Programmierung, Objekte, Methoden, Ereignisse
Oberflächengestaltung		
Musik: Trommelrhythmus	Rhythmus-Mikrowelt	Befehle, Objekte
Deutsch: Rittergeschichte		Klasse <i>Textfeld</i> , <i>Igel</i>
Automatisierung		
Allgemein: Animation von Objekten		Objektorientierte Programmierung, Namensgebung, Objekte, Methoden
Syntax, Semantik		
Deutsch: Generierte Gedichte	Satz-Mikrowelt	Syntaxdiagramme, Funktionen, Maschinen, zusammengesetzte Maschinen, Oberflächengestaltung, Klasse <i>Schaltfläche</i> , objektorientierte Programmierung, Ereignisse

Tabelle 4.2: Informatische Grundkonzepte in der Projektphase

Modularisierung

Mathematischer Inhalt der Rittergeschichte war die Achsenspiegelung. Die Schülerinnen und Schüler erstellten einen Befehl `Viertel.Burg`, mit dem der Igel die in Abbildung 4.34 dargestellte Viertel-Burg zeichnete.

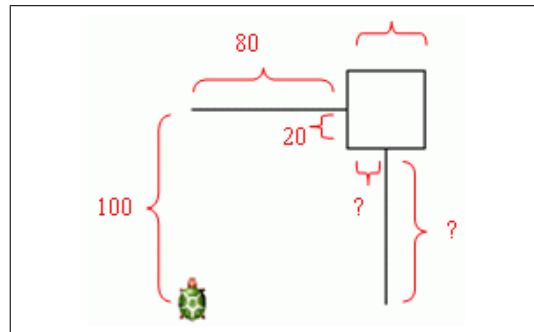


Abbildung 4.34: Viertel-Burg

Anschließend führten sie zunächst Spiegelungen mit Papier und Bleistift durch, um aus dem Viertel die gesamte Burg zu konstruieren. Sie erkannten, dass insgesamt drei Spiegelungen notwendig waren und beschrieben die Spiegelachsen in Abhängigkeit der Blickrichtung des Igels. Dies erleichterte ihnen die Analogiebildung zu den eingeführten Befehlen

- `spiegel_Blickrichtung`
- `spiegel_senkrecht_zur_Blickrichtung`
- `spiegel_Blickrichtung_und_Senkrechte`

der Spiegel-Mikrowelt. Sie erhielten im Eingabeparameter den Namen des Befehls übergeben, in dem der Igel die zu spiegelnde Grundfigur zeichnete. Deutlich wurde bei der Arbeit, dass in modularen Umgebungen gewisse Regeln bei der Erstellung der Module eingehalten werden müssen. Da die Spiegelachsen der drei Befehle abhängig von der Blickrichtung und Position des Igels am Ende des Zeichenprozesses waren, musste der Igel in **Viertel_Burg** am Ende wieder in seinen Ausgangszustand gebracht werden, d.h. er musste an der gleichen Position stehen und in die gleiche Richtung schauen.

Netzwerk

In Englisch wurde auf einer Seite der multimedialen Geschichte der Stammbaum der Ritterfamilie als Netzwerk modelliert und visualisiert. Hier brachten sich die Schülerinnen und Schüler vor allem kreativ bei der Darstellung des Stammbaums ein.

Objektorientierung

Zu dem Stammbaum erstellten die Kinder ein interaktives Rätsel. Ja-/Nein-Fragen zu verwandtschaftlichen Beziehungen formulierten sie auf Englisch mit Hilfe des Genitivs. Für jede Frage stellten sie zwei Antwortschaltflächen zur Verfügung. Richtig beantwortete Fragen färbten sie grün und falsche rot. Zur Initialisierung des Rätsels programmierten sie ebenfalls eine Schaltfläche.

Insgesamt gab es zahlreiche Objekte, die miteinander programmiertechnisch verknüpft waren. Aus diesem Grund wurden die Schülerinnen und Schüler mehrfach aufgefordert,

Is madam Nico Ugala Ugala-Ugala's father ?

Is Tobias Ugala madam Ugala-Ugala's father?

Is Floh Ugala madam Ugala-Ugala's mother?

Solution: **wrong**

Abbildung 4.35: Interaktives Rätsel zum Stammbaum der Ritterfamilie

den Objekten sinnvolle Bezeichnungen wie `text_question1` oder `question1_button_no` zu geben, damit diese im Programmcode leichter angesprochen werden konnten. Die Methoden der einzelnen Objekte wurden nicht mehr über das Kontextmenü mit der Maus ausgewählt, sondern abstrakt durch Anweisungen wie

```
text_question1.setBGColour "green"
```

formuliert. Die fast natürlichsprachliche Anweisung verdeutlicht die Struktur der Objektauswahl, die Zuordnung der Methode zu dem Objekt und die Spezifikation des Attributwerts.

Oberflächenmodellierung

Bei der Erstellung der Trommelrhythmen spielte keines der informatischen Grundkonzepte explizit eine Rolle. Vielmehr wurde die Rhythmus-Mikrowelt, die acht verschiedene Töne zur Verfügung stellte, einfach angewendet (vgl. Abbildung 4.36). Allerdings war es nicht möglich, Rhythmen mit Hilfe einer Schaltfläche abzuspeichern, sondern nur der aktuell dargestellte Rhythmus konnte abgespielt werden. Um Rhythmen auf anderen Seiten des Projekts einzubauen, mussten die Schülerinnen und Schüler ihre Kompositionen mit dem Befehl `speicher_Rhythmus` abspeichern. Mit `spiele_Rhythmus` konnten diese dann an beliebiger Stelle eingebaut werden. Die Oberfläche der Mikrowelt war also nur begrenzt komfortabel zu bedienen.

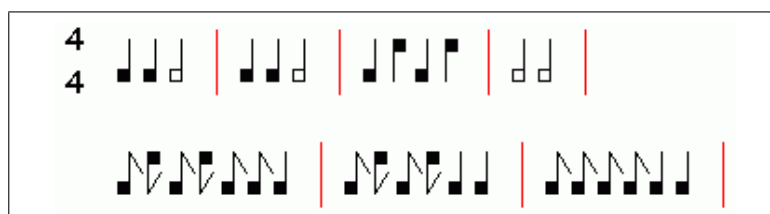


Abbildung 4.36: Rhythmus eines Schülerpaares

Automatisierung

Fast alle Schülerinnen und Schüler animierten auf mindestens einer Seite eine ihrer zahlreichen Figuren. Die Bewegungen automatisierten sie mit Hilfe des Befehls **FOREVER** entweder mit konstanter Geschwindigkeit auf einer Geraden oder zufällig. Auch bei dieser Arbeit vertieften sie Kenntnisse der objektorientierten Programmierung, da wie beim Rätsel zum Stammbaum der Ritterfamilie die Methoden von Objekten im Programmcode aufgerufen wurden. Eine Gruppe bewegte zum Beispiel ihren Drachen folgendermaßen über den Bildschirm:

```
dragon'FOREVER [FORWARD 20  RIGHT 80  WAIT 100  
                FORWARD 20  RIGHT 60  WAIT 100]
```

Syntax, Semantik

Im Fach Deutsch ließen die Schülerinnen und Schüler das Programm ein Gedicht generieren. Thematisch vorgegeben war eigentlich ein Liebesgedicht des Ritters an sein Ritterfräulein. Allerdings erwies sich das Thema als zu heikel. Stattdessen wurde an einem Vers, bestehend aus Subjekt und Prädikat, gearbeitet. Der syntaktische Aufbau der Verse wurde mit einem Syntaxdiagramm an der Tafel dargestellt und die Klasse suchte gemeinsam entsprechende Beispiele. Die Bestandteile der Verse setzten die Schülerinnen und Schüler in Funktionen um, die sie zur automatischen Generierung der Verse nutzten. Anschließend entwarfen sie auf einer Seite des Projekts eine Benutzerschnittstelle, mit deren Hilfe das Gedicht am Bildschirm ausgegeben werden konnte. Im Ereignis **onPush** der Schaltfläche hinterlegten sie den entsprechenden Code. Beim Testen fiel ihnen allerdings auf, dass die Verse übereinander geschrieben wurden. Als Lösung setzten sie den schreibenden Igel nach der Ausgabe des Verses entweder direkt in eine neue Zeile oder fügten eine neue Schaltfläche zur Bewegung des Igels ein. Eine sehr kreative Gruppe drehte den Igel nach jedem Vers, so dass ein Gedicht in Form von Sonnenstrahlen entstand.

4.3 Fachunabhängiger Unterricht

Für einige informatische Grundkonzepte konnten nur sehr schwierig fachliche Kontexte gefunden werden. Aus diesem Grund waren am Ende des Schuljahrs die Modellierung, Objektorientierung, Modularisierung, Hierarchisierung und Rekursion im Unterricht noch überhaupt nicht oder nicht intensiv behandelt worden. Bis auf letztere können alle dem *Programmieren im Großen* zugeordnet werden, das am besten in umfangreichen Programmierprojekten thematisiert werden kann. Obwohl sich die Schülerinnen und Schüler über mehrere Monate sehr intensiv mit der Logo-Programmierung auseinandergesetzt hatten, hätte sie ein großes, eigenständiges Programmierprojekt überfordert. Aus diesem Grund wurde in diesem letzten Teil des Unterrichtsversuchs die Erweiterung be-

reits existierender Logo-Programme als methodische Form gewählt. Dieses Vorgehen bot sich an, da die grundlegenden Strukturen der Programme erarbeitet werden mussten und dabei die Prozesse der Modularisierung, Hierarchisierung und Modellierung nachvollzogen werden konnten. Als Themen wurden zwei Spiele und ein Malprogramm gewählt, da Humbert (2001) fordert, dass die Problemstellungen, die im Unterricht behandelt werden, nicht nur fachlich zu begründen sind, sondern sich auch an den Interessen der Schülerinnen und Schüler orientieren müssen.

Neue informatische Grundkonzepte		
Fachinhalt	Mikrowelt	Wiederholung
Modellierung		
Erweiterung eines Autospiels	Autospiel-Mikrowelt	Modularisierung, Hierarchisierung, Objektorientierung, Ereignisse, Oberflächengestaltung
Rekursive Struktur		
Englisch: Wortfeld Kleider im Windows-Explorer		Verzeichnis, Datei, Dokument, Hierarchie
Beschreibung rekursiver Figuren		Rekursive Strukturen erkennen, beschreiben und zeichnen
Modellierung		
Entwurf eines Malprogramms und Erstellung eines Labyrinthspiels	Mal-Mikrowelt, Labyrinth-Mikrowelt	Objektorientierung, Oberflächengestaltung

Tabelle 4.3: Informatische Grundkonzepte ohne Fachbezug

Modellierung

Die Unterrichtseinheit *Autospiel* wurde konzipiert, um die Lernenden in Kontakt mit der Modularisierung, Hierarchisierung, Objektorientierung und Modellierung zu bringen, ohne gleichzeitig ihre programmiertechnischen Fähigkeiten zu überfordern. In Abbildung 4.37 ist das Autospiel in seinem Anfangszustand dargestellt. Der Rennwagen, der sich mit konstanter Geschwindigkeit bewegte, musste auf der vorgegebenen Fahrbahn gehalten werden. Der Wagen konnte mit den Pfeiltasten in die vier Himmelsrichtungen gelenkt werden. Relative Richtungsänderungen waren nicht möglich. Am Rande der Fahrbahn befanden sich Objekte, die beim Überschreiten von bestimmten Streckenabschnitten zu Hindernissen auf der Fahrbahn wurden. Die Ereignisse, die die Hindernisse verursachten, wurden mit Hilfe von Geräuschen dargestellt. Ein Sturm ließ zum Beispiel die Bäume umstürzen und ein heftiger Regen den Teich anschwellen. Mit Hilfe der Schaltfläche *Neues Autospiel* konnten der Rennwagen auf seine Startposition gesetzt und alle Hindernisse von der Fahrbahn entfernt werden. Mit *Start* wurde der Motor des Wagens gestartet und die kontinuierliche Rennfahrt begonnen.

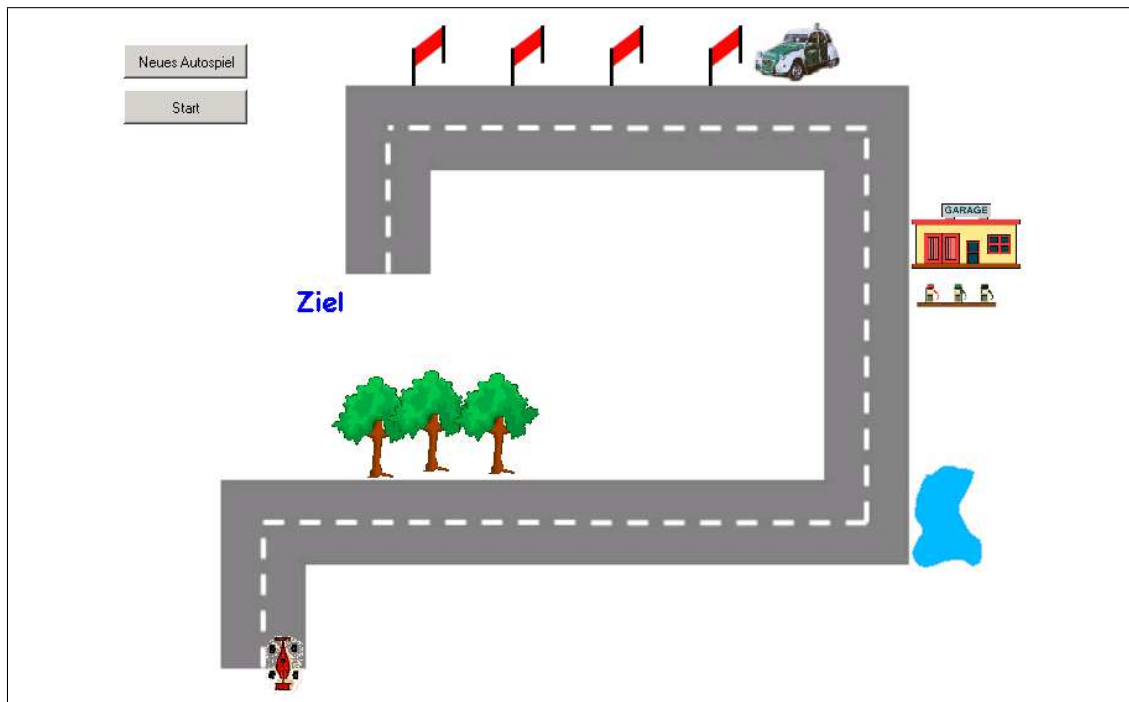


Abbildung 4.37: Benutzeroberfläche des Autospiels

In Wursthorn (2005a, S. 317) wird beschrieben, wie das Erlernen der informatischen Grundkonzepte mit dem Spielen am Computer verknüpft werden muss, um aufgebautes Wissen anschließend beim konstruktiven Arbeiten zu festigen und zu vertiefen. Dort werden drei Arbeitsformen beschrieben:

- Spielen
- Analysieren
- Verändern.

In der Spielphase steht das zweckfreie Spielen im Vordergrund, um die Funktionalität kennen zu lernen. Daran schließt sich eine Analysephase, in der der innere Aufbau des Programms durch Codestudium erarbeitet und in der anschließenden dritten Phase aktiv verändert wird. Schon bei der Konzeption des Spiels muss darauf geachtet werden, dass sinnvolle Veränderungs- und Erweiterungsmöglichkeiten enthalten sind, damit die Schülerinnen und Schüler möglichst selbstständig arbeiten können.

Beim Autospiel war dies auf drei verschiedene Arten möglich. Zunächst konnten neue Spiel-Modi eingeführt werden, die das Rennauto in unterschiedlichen Geschwindigkeiten über die Fahrbahn bewegten (siehe Zeile 14 im Programmcode). Dies hatte zur Konsequenz, dass die Kommunikationsschnittstelle zur Auswahl der verschiedenen Modi angepasst werden musste. Eine zweite Veränderungsmöglichkeit bestand in der Verwendung anderer Steuerungstasten (vgl. Zeile 5 – 8 im Programmcode). Zuletzt erschien es noch sinnvoll, die absoluten Richtungsänderungen durch relative zu ersetzen, so dass das

Fahrzeug auch auf einer kreisförmigen Bahn mit Hilfe von zwei Tasten gelenkt werden konnte (siehe ebenfalls Zeile 5 – 8).

```
1:  to Starte_Autofahrt
2:
3:    Auto'setKeyMenu [
4:      esc    [cancel "move]
5:      up     [Auto'setHeading 0]
6:      right  [Auto'setHeading 90]
7:      down   [Auto'setHeading 180]
8:      left   [Auto'setHeading 270]
9:    ]
10:
11:    playWave ".\\Spiel\\Sound\\AutoAnlasser.WAV
12:    wait 3000
13:
14:    (Auto'forever [fd 1 wait 10
15:      ... ]
16:    )
17:  end
```

Die ersten beiden Phasen wurden sehr stark mit Hilfe von Arbeitsblättern gelenkt. Es mussten die Stellen auf der Fahrbahn, an denen die Hindernisse entstanden, markiert, die Objekte an der Programmoberfläche benannt und die Befehlsstrukturen des gesamten Programms erarbeitet werden. Dies stellte sicher, dass nicht nur gespielt, sondern auch Wissen über den Programmaufbau erworben wurde.

Obwohl drei Alternativen zur Veränderung des Spiels vorgegeben waren, arbeiteten alle Gruppen an neuen Spielmodi. Sie fügten meist mehrere neue Geschwindigkeitsstufen an der Benutzerschnittstelle ein und implementierten für diese neue Befehle. Dabei erweiterten sie softwareergonomisch ansprechend die Benutzerschnittstelle und passten das Modell des Autospiels an.

Die Unterrichtseinheit machte den Schülerinnen und Schülern sehr viel Spaß, obwohl sie an den sehr anspruchsvollen informatischen Konzepten der Modularisierung, Hierarchisierung, Objektorientierung, Modellierung und Softwareergonomie arbeiteten und sich in fremden Programmcode einlesen mussten. Besonders die Möglichkeit spielen und eigenständige „Verbesserungen“ in das bereits bestehende Programm einfügen zu können, waren für die hohe Motivation verantwortlich.

Rekursion

Zur Einführung von rekursiven Vorstellungen wurde nicht mit Logo, sondern mit dem Windows-Explorer gearbeitet. Auf einem Arbeitsblatt ordneten die Schülerinnen und Schüler englische Begriffe aus dem Wortfeld *Kleider* in eine vorgegebene Hierarchie

ein. Außerdem lagen sowohl die Blätter als auch die inneren Knoten der Hierarchie im Windows-Explorer elektronisch als Bilder in Dateien und als Verzeichnisse vor, die hierarchisch sortiert werden mussten. Die Beziehungen zwischen Daten, Dateien und Verzeichnissen wurden in Form der Metaphern *beschriftete Schachtel* und *beschrifteter Schrank* in einer Frontalphase vorgestellt. Die rekursive Struktur der Verzeichnisse wurde durch das Bild *in Schränken können sich wieder Schränke befinden* veranschaulicht.

Die Schülerinnen und Schüler arbeiteten in dieser Unterrichtseinheit sehr konzentriert, da ihnen die Vorstellung von Dateien und Verzeichnissen als Schachteln und Schränke gefiel und sie Erfolgserlebnisse beim Einsortieren der Begriffe in die Hierarchie hatten. Auch die anschließende kreative Phase, in der sie mit den Bildern ein Kleidergeschäft einrichteten, wurde sehr positiv bewertet.

Explizit wurde die Rekursion erst beim Zeichnen von Figuren eingeführt. Zunächst wurde eine rekursive Figur an die Tafel gemalt und auf Basis der Beschreibung der Schülerinnen und Schüler erneut gezeichnet. Aufgrund von Ungenauigkeiten entstand nicht die Ausgangsfigur. Anschließend mussten die Schülerinnen und Schüler selbst nach der Anweisung

„Achtrekeln“ bedeutet, in dem oberen Kreis der Acht eine neue Acht zu zeichnen und diese neue Acht dann zu „Achtrekeln“

die Figur vervollständigen.

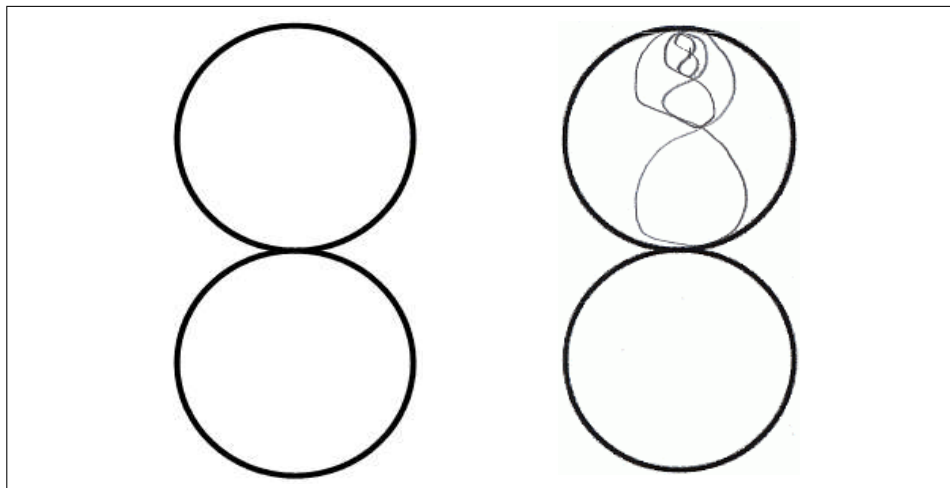


Abbildung 4.38: Rekursive Figur – Vorlage und Lösung einer Schülerin

Gleiches wurde noch an zwei weiteren Beispielen geübt. Aufgrund der Misserfolge bei der Beschreibung der Figuren an der Tafel, arbeiteten die Schülerinnen und Schüler sehr zögerlich und ließen sich ständig die Richtigkeit ihrer Lösungen bestätigen. Besonderen Spaß machte ihnen das Zeichnen einer eigenen rekursiven Figur. Bei der abschließenden

Beschreibung vorgegebener rekursiver Figuren griffen viele jedoch wieder auf iterative Formulierungen zurück (siehe Kapitel 6.4.3).

Modellierung

Um auch denjenigen Schülerinnen und Schülern, die nicht die Computer AG besuchten, die Möglichkeit anzubieten, auf den Convertibels mit dem Stift zu arbeiten und um die Konzepte der Modellierung von Benutzeroberflächen zu vertiefen, wurde ein Malprogramm mit einer Farbe und Strichstärke zur Verfügung gestellt. Nach dem Drücken der Schaltfläche *Starte Malprogramm* konnte mit dem Stift gezeichnet werden. Es gab die Möglichkeit, die Zeichnung auf dem Bildschirm zu löschen, das Bild in einer Datei abzuspeichern oder ein bereits bestehendes Bild zu laden. Mit *Beende Malprogramm* wurde der Malmodus verlassen. Die Aufgabe der Kinder bestand in der selbstständigen Erweiterung des Programms um weitere Farben und Strichstärken.

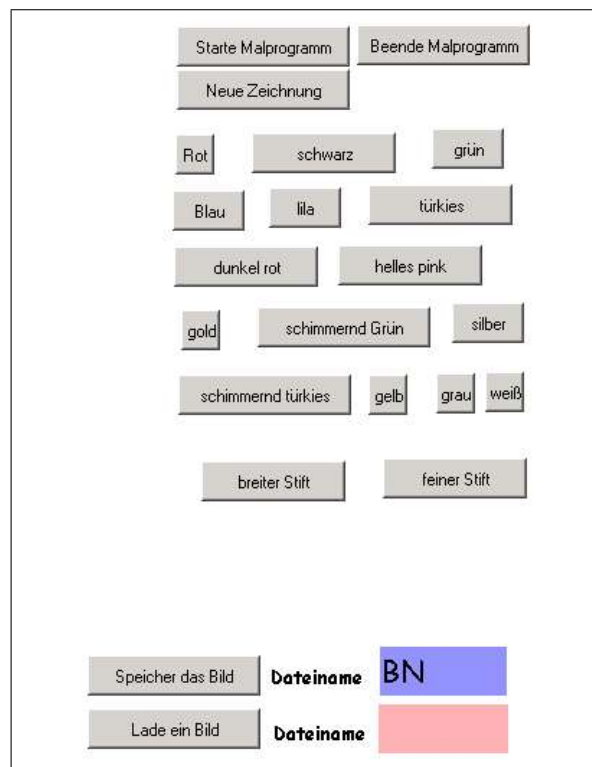


Abbildung 4.39: Benutzeroberfläche des Malprogramms eines Schülerinnenpaares

Es wurde ein Beispiel an der Tafel mit dem Hinweis besprochen, im Ereignis **onPush** der beiden Schaltflächen *Rot* und *feiner Stift* die Lösung nachschauen zu können. Die verschiedenen Codes der Farben und Strichstärken wurden auf zwei weiteren Seiten der Mal-Mikrowelt angegeben, so dass ohne Arbeitsblätter gearbeitet werden konnte. Die Vorgehensweise der Schülerinnen und Schüler unterschied sich sehr. Die einen waren bestrebt, ein komplettes Malprogramm mit möglichst vielen Farben und unterschiedlichen

Strichstärken (siehe Abbildung 4.39) zu erzeugen und andere priorisierten das Zeichnen mit dem Stift. Alle waren motiviert, da sie mit dem Stift als Eingabegeräte arbeiten durften. Gleichzeitig vertieften sie die objektorientierten Konzepte bei der Programmierung der Schaltflächen und modellierten Benutzerschnittstellen. In einem zweiten Schritt musste ein Labyrinthspiel nach vorgegebenen Schritten vervollständigt werden. Ziel war es, in der Mal-Mikrowelt ein Labyrinth zu zeichnen, es dort abzuspeichern und dann in der Labyrinth-Mikrowelt zu laden, um dort mit einem Mäuschen zu spielen. Durch den auf dem Arbeitsblatt vorgegebenen Programmcode

```
if equal? mouse'dotColour "red [
```

wurde gezeigt, wie das Mäuschen feststellen konnte, ob es gegen eine Wand des Labyrinths lief. Dies verdeutlichte den Kindern, dass sie ihr Labyrinth in der Mal-Mikrowelt nur in einer Farbe zeichnen durften und diese dann in der Bedingung des Befehls **starte_Labyrinthspiel** angeben mussten. Das Modell des Programms hatte also Schwächen, die den Schülerinnen und Schülern beim weiteren Experimentieren auffielen.

Einige Schülerinnen und Schüler waren nicht motiviert genug, die Arbeit an dem Malprogramm aufzugeben und sich mit dem Labyrinth auseinanderzusetzen. Ihnen war es zu mühsam, sich die vier Arbeitsschritte

- Labyrinth im Malprogramm speichern,
- Malprogramm schließen,
- Labyrinthspiel starten und
- im Malprogramm abgespeichertes Labyrinth laden

zu verdeutlichen, um überhaupt das Labyrinthspiel nutzen zu können. So lagen nur sehr wenige lauffähige Labyrinthspiele vor. Trotzdem ist es wichtig, Aktivitäten dieser Art anzubieten, da Kinder auch aus verschiedenartigen Anwendungen auswählen, manche ablehnen und andere vertiefend durchführen können sollen.

5 Konzeption und Durchführung der Untersuchung

5.1 Fragestellung und Untersuchungsdesign

Im Zentrum dieser Untersuchung steht die Frage nach der Eignung des Unterrichtskonzepts zur Vermittlung der informatischen Grundkonzepte in Klassenstufe 5 der Realschule. Zu berücksichtigen sind das kognitive Erlernen der informatischen Grundkonzepte, die Beurteilung des Unterrichts durch die Lehrerinnen, Schülerinnen und Schüler, die im Unterricht entstandenen Arbeitsergebnisse und die Einstellungen der Schülerinnen und Schüler bezüglich der Verwendung des Computers.

Daraus ergeben sich für die Auswertung folgende zentrale Fragestellungen:

- Eignet sich das Gesamtkonzept zur Vermittlung der informatischen Grundkonzepte in Klassenstufe 5 der Realschule?
- Lässt sich ein Lernerfolg für die informatischen Grundkonzepte feststellen?
- Gibt es beim Erlernen der informatischen Grundkonzepte Unterschiede bezüglich des Geschlechts, des kognitiven Gesamtleistungsniveaus oder der sprachgebundenen, zahlengebundenen oder formallogischen Fähigkeiten?
- Wie verändern sich die Einstellungen der Schülerinnen und Schüler bezüglich des Computers?

Für die Untersuchung dieser Fragestellungen wurden geeignete Erhebungsinstrumente ausgewählt oder entwickelt. Vor allem qualitative Methoden sind geeignet, um die verschiedenen Aspekte des Unterrichtsversuchs zu berücksichtigen. Aus diesem Grund wurde sowohl mit den Lehrerinnen als auch mit sechs Schülerinnen und Schülern am Ende des Schuljahrs ein Leitfadeninterview durchgeführt und von der Forscherin selbst ein Forschungstagebuch über die einzelnen Unterrichtsstunden geschrieben. Außerdem existieren Unterrichtsmaterialien wie Arbeitsblätter und Logo-Programme, die qualitativ ausgewertet wurden.

Quantitative Instrumente wurden zur Ermittlung des Lernerfolgs der informatischen Grundkonzepte, der Erfassung des kognitiven Gesamtleistungsniveaus, des sprachgebundenen, zahlengebundenen und formallogischen Denkens und zur Messung der Einstellungen zum Computer verwendet.

Die zeitliche Abfolge des Einsatzes der verschiedenen Erhebungsinstrumente wird im

Untersuchungsplan in Abbildung 5.1¹ skizziert.

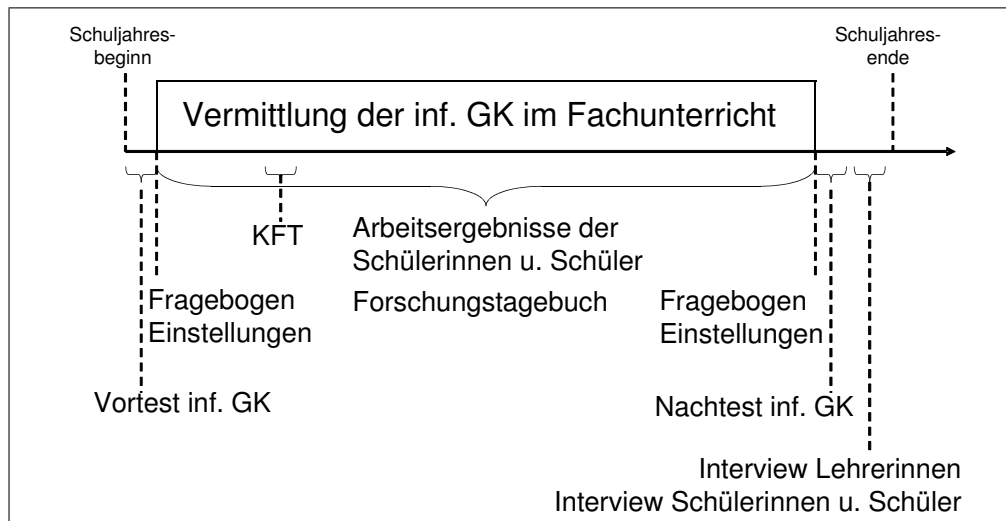


Abbildung 5.1: Untersuchungsplan

5.2 Quantitative Erhebung

Um Aussagen über den Lernerfolg der informatischen Grundkonzepte Algorithmisierung, Automatisierung, Codierung, Datenstrukturen, Modellierung, Objektorientierung, Modularisierung, Hierarchisierung, Abstraktion, Iteration, Rekursion, Programmablaufpläne, Syntaxdiagramme, UML-Klassendiagramme, Hierarchien, Netzwerke, Grammatik, Syntax, Semantik und Funktionen ganz allgemein und in Abhängigkeit der Fähigkeiten der Schülerinnen und Schüler machen zu können, muss das Erlernen dieser Konzepte quantifiziert werden. Da es für die informatischen Grundkonzepte keine standardisierte Form der Operationalisierung gibt, musste ein eigenes Erhebungsinstrument entwickelt werden (vgl. Kapitel 5.2.1). Zur Erfassung des kognitiven Gesamtleistungsniveaus und der sprachgebundenen, zahlengebundenen und formallogischen Fähigkeiten der Schülerinnen und Schüler wurde der KFT (*Kognitiver Fähigkeitstest*) (Heller & al., 2000) eingesetzt. Es handelt sich um einen differentiellen Intelligenztest zur Ermittlung der kognitiven Ausstattung von Schülerinnen und Schülern. Er liefert Informationen über sprachliches, quantitatives und nonverbal-figurales Denken. Außerdem wurden noch die Veränderungen der Einstellungen der Schülerinnen und Schüler bezüglich des Computers als Arbeitsgerät mit Hilfe eines selbst entwickelten Fragebogens zu Beginn und am Ende des Unterrichtsversuchs gemessen.

¹KFT – Kognitiver Fähigkeitstest; inf. GK – informatische Grundkonzepte

Im Folgenden werden zunächst die Entwicklung des Tests zu den informatischen Grundkonzepten skizziert und die Testgütekriterien diskutiert, bevor das Ausgangsmaterial und die Auswertungsverfahren vorgestellt werden.

5.2.1 Testentwicklung zu informatischen Grundkonzepten

Da Schülerinnen und Schüler zu Beginn der Klassenstufe 5 möglicherweise informatische Grundkonzepte anwenden, jedoch begrifflich nicht fassen können, muss ein Leistungstest genau dieses Anwendungswissen in der Erfahrungswelt der Kinder überprüfen. Die Testitems müssen also ohne Verwendung der Begriffe der informatischen Grundkonzepte auskommen. Ein Item zur Modularisierung sah zum Beispiel folgendermaßen aus:

Aufgabe:

Das folgende Bild wurde mit Kartoffelstempeln erstellt. Beschreibe genau, wie 8 Personen das Bild möglichst schnell herstellen können. Es müssen die Kartoffelstempel geschnitzt und das Bild mit den Stempeln gedruckt werden.

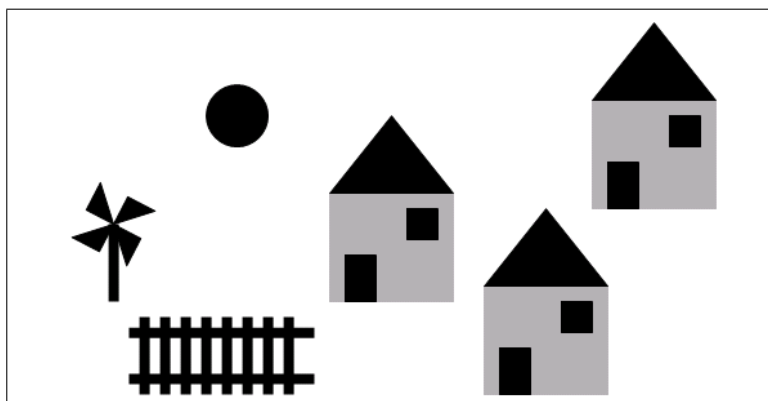


Abbildung 5.2: Aufgabe zur Modularisierung

Insgesamt enthält das Bild acht verschiedene geometrische Figuren, so dass acht Kartoffelstempel hergestellt werden müssen. Die Arbeit wird dann effizient aufgeteilt, wenn jede Person genau einen Stempel schnitzt und somit einen unabhängigen Grundbaustein für die Gesamtlösung zur Verfügung stellt. Für die Lösung der Aufgabe ist also nur Anwendungs- und kein Begriffswissen entscheidend.

In einem ersten Schritt wurden von der Forscherin zu allen informatischen Grundkonzepten Aufgaben entwickelt und mit Kollegen in einem Expertenteam wiederholt diskutiert und erweitert. Dabei wurde deutlich, dass das Verständnis gewisser Konzepte, wie zum Beispiel die Modellierung, nicht in Form eines schriftlichen Leistungstests gemessen werden kann.

Als Pilotuntersuchung wurde der Test am Ende des Schuljahrs 2002/2003 in einer 5. Klasse durchgeführt. Die Schülerinnen und Schüler erhielten die kompletten Testaufgaben mit der Arbeitsanweisung, in 45 Minuten beliebig viele Aufgaben zu lösen. Schon während der Aufgabenbearbeitung wurde deutlich, dass die Kinder größtenteils durch die vielen Aufgabenblätter irrten und die einzelnen Aufgaben nur oberflächlich bearbeiteten. Mit der selbstständigen Zeiteinteilung waren sie überfordert. Ein Tag später führte die Forscherin mit einigen Kindern ein kurzes Interview, um mögliche Schwierigkeiten zu erfahren. Es bestätigte sich, dass Aufgaben teilweise überhaupt nicht richtig gelesen und Formulierungen nicht verstanden worden waren. Die Testitems wurden daraufhin sprachlich angepasst, Erklärungen ergänzt, Darstellungen verändert und die Anzahl der Aufgaben gekürzt, so dass alle Testitems in 50 Minuten bearbeitet werden konnten. Auch die Durchführung des Tests wurde angepasst. In der zweiten Pilotstudie in einer weiteren 5. Klasse las die Forscherin die einzelnen Aufgaben vor. Die Schülerinnen und Schüler bearbeiteten sie direkt im Anschluss. Das Vorgehen bewährte sich, so dass dieses beibehalten wurde. Vereinzelt änderten sich nach der Auswertung der zweiten Pilotstudie noch Testitems, doch insgesamt erwies sich die Version als sehr stabil und konnte somit in der Versuchsklasse eingesetzt werden.

Der Gesamttest wurde in drei gleich große Teile zerlegt. Jeder wurde mit acht Punkten bewertet und konnte in einer Schulstunde bearbeitet werden. Da der Test zu Beginn und am Ende des Unterrichtsversuchs durchgeführt wurde, um den Lernerfolg als Differenz der Testergebnisse zu messen, wurde eine zweite leicht modifizierte Form entwickelt, um die Wiedererkennung von Aufgabenstellungen zu vermeiden. Die Tabellen 5.1, 5.2 und 5.3 geben einen Überblick über die einzelnen Testitems der drei Testteile und die dort geprüften informatischen Grundkonzepte.

Aufgaben	Informatische Grundkonzepte
Lebewesen auf Plauturus	Objektorientierung
Einordnung von Begriffen	Objektorientierung, Hierarchie
Ordnung von Figuren	Objektorientierung, Attribute
Kreisdiagramme	Objektorientierung, Attribute
Folgen	Iteration
„Quadrekel“	Rekursion, Algorithmus
Zeichnung	Iteration, Algorithmus
Beschreibung	Iteration, Rekursion, Algorithmus

Tabelle 5.1: Aufgaben zu den informatischen Grundkonzepten (1. Teil)

Beurteilung der Testgüte

Den drei zentralen Kriterien der Testgüte *Objektivität*, *Reliabilität* und *Validität* kann die endgültige Version des Tests zu den informatischen Grundkonzepten nicht standhalten. In Bortz und Döring (2002, S. 194) wird die Objektivität als ein Maß definiert, das

Aufgaben	Informatische Grundkonzepte
Bild gemeinsam zeichnen	Problemlösestrategie, Modularisierung
Kartoffelstempel	Problemlösestrategie, Modularisierung
Wunderstempel	Problemlösestrategie, Abstraktion
Eigene Adresse	
Adressen anordnen	Datenstruktur
Fragen zu Adressen beantworten	
Adressanordnung	Datenstruktur
Rezept	Datenstruktur
Variables Rezept	Abstraktion
Morgenqualen	Algorithmus, Wissensrepräsentation, Programmablaufplan

Tabelle 5.2: Aufgaben zu den informatischen Grundkonzepten (2. Teil)

Aufgaben	Informatische Grundkonzepte
Formen im Land „Droblavie“	Sprache, Grammatik
„Formalien“	Sprache, Grammatik
Roboter	Sprache, Grammatik
Geheimschrift	Dekodierung
Eigene Geheimschrift	Codierung
1. Steigerungsform bilden	
Steigerungsform „Motor“	Wissensrepräsentation, Funktion
Bildung von Sätzen	Wissensrepräsentation, Syntaxdiagramm
Unregelmäßige Steigerungsformen	Wissensrepräsentation, Programmablaufplan

Tabelle 5.3: Aufgaben zu den informatischen Grundkonzepten (3. Teil)

angibt, wie stark die Testergebnisse vom Testanwender unabhängig sind. Die Kriterien der Durchführungs- und Auswertungsobjektivität sind durch die genaue Beschreibung des Vorgehens (vgl. Kapitel 5.2.2) erfüllt.

Die Zuverlässigkeit und die Gültigkeit des Tests konnten aus Zeitgründen nicht anhand von Reliabilitäts- und Validitätsprüfungen gemessen werden. Hier sind in Zukunft weitergehende umfassende Forschungsarbeiten notwendig. Die Testergebnisse können trotzdem als Indikatoren für den Lernerfolg bezüglich der informatischen Grundkonzepte interpretiert werden. Abgesichert werden sie durch die aus den qualitativen Auswertungen der Interviewdaten und Unterrichtsmaterialien abgeleiteten Aussagen.

5.2.2 Test zu informatischen Grundkonzepten

Stichprobe

Am Vor- und Nachtest zu den informatischen Grundkonzepten sowohl zu Beginn als auch am Ende des Schuljahrs nahmen alle Schülerinnen und Schüler der Versuchsklasse teil.

Material

Der Leistungstest bestand aus drei Testteilen, die jeweils in einer Unterrichtsstunde bearbeitet wurden. Der Vortest des Leistungstests ist in Anhang B abgebildet.

Durchführung

Die drei Testteile wurden an drei verschiedenen Tagen im Klassenverband bearbeitet. Da nur eine Form des Tests vorlag, wurde die Klasse geteilt, so dass gegenseitiges Abschreiben ausgeschlossen werden konnte. Jede Aufgabe wurde von der Forscherin genau zweimal vorgelesen und direkt im Anschluss bearbeitet. Die Kinder, die einen Testteil wegen Erkrankung versäumten, schrieben diesen möglichst zeitnah nach. Der Test selbst wurde in der Klasse nicht besprochen, sondern es wurden nur die Testergebnisse bekannt gegeben.

Auswertung

Die Aufgaben des Tests wurden von der Forscherin einzeln ausgewertet. Nur vollkommen richtige Ergebnisse wurden mit einem Punkt bewertet. Insgesamt konnten maximal 24 Punkte erreicht werden.

5.2.3 Kognitiver Fähigkeitstest

Stichprobe

Alle Schülerinnen und Schüler der Versuchsklasse nahmen am KFT (*Kognitiver Fähigkeitstest*) (Heller & al., 2000) teil.

Material

Der KFT wurde eingesetzt, um Differenzen bezüglich des kognitiven Gesamtleistungsniveaus und der sprachgebundenen, zahlengebundenen und formallogischen Fähigkeiten zu messen. Die drei Testteile des KFT bestanden jeweils aus drei Aufgabenreihen. Der Verbal-Teil setzte sich aus den Subtests *Wortschatz*, *Wortklassifikation* und *Wortanalogien*, der Quantitativ-Teil aus *Mengenvergleichen*, *Zahlenreihen* und *Gleichungenbilden* und der Nonverbal-Figural-Teil aus *Figurenklassifikationen*, *Figurenanalogien* und *Faltaufgaben* zusammen. Die Ergebnisse aller drei Testteile beschrieben das kognitive Ge-

samtleistungsniveau. Dies bezeichnet Heller & al. (2000, S. 46) als Maß für die allgemeine Intelligenz.

Durchführung

Der KFT wurde vor den Herbstferien nach ca. 6 Wochen Unterricht als Gruppentest mit zwei parallelen Formen durchgeführt. Die Klasse war geteilt, so dass in einer Gruppengröße von 15 Kindern gearbeitet wurde. Diejenigen, die einen Testteil verpassten, holten ihn in einer Einzelsitzung nach.

Auswertung

Der KFT wurde mit Hilfe der vorgegebenen Auswertungsschablonen evaluiert. Nach den Anweisungen wurden die Werte der Subtests in jedem Testteil addiert, die Gesamtleistung als deren Summe gebildet und die resultierenden Werte mit der beigefügten Normentabelle in den für Klasse 5 schultypischen Standard transformiert.

5.2.4 Fragebogen zu Einstellungen zum Computer

Stichprobe

Sowohl in der zweiten Woche des Schuljahrs als auch in der drittletzten Schulwoche erhielten alle 30 Schülerinnen und Schüler einen Fragebogen zu den Einstellungen zum Computer. Der Rücklauf des Fragebogens zu Beginn des Schuljahrs war vollständig. Am Ende des Schuljahrs fehlten zwei Bögen. Außerdem waren nicht alle Items bearbeitet. Die Antworthäufigkeiten der einzelnen Items lag zwischen 21 und 30. Die genaue Verteilung ist in Anhang D dargestellt.

Material

Zur Erfassung der Veränderungen der Einstellungen zum Computer als Arbeitsgerät wurde ein selbst entwickelter Fragebogen mit 39 Items zu den Inhaltsbereichen

- motivationale Aspekte bei der Verwendung des Computers (9 Items)
- Gefühle bei der Benutzung des Computers (15 Items)
- Selbsteinschätzung der Kompetenzen im Umgang mit dem Computer (5 Items)
- Beurteilung der Eigenschaften des Computers (10 Items)

eingesetzt. Die Items mussten auf einer 5-stufigen Rating-Skala, die mit den verbalen Marken *stimmt überhaupt nicht*, *stimmt eher nicht*, *teils teils*, *stimmt eher* und *stimmt voll* gekennzeichnet war, beurteilt werden. Der Fragebogen ist in Anhang C abgebildet.

Durchführung

Die Bearbeitung des Fragebogens erfolgte als Hausaufgabe. Besonders am Ende des Schuljahrs erfolgte die Rückgabe der ausgefüllten Fragebögen nicht von allen Kindern

am vorgegebenen Termin, so dass manche einen etwas längeren Bearbeitungszeitraum hatten.

Auswertung

Für jedes Item wurden sowohl für die Vor- als auch Nachbefragung die zentralen Tendenzen der Einschätzungen der Klasse mit dem Median und dem ersten (Q1) und dritten Quartil (Q3), die die mittleren 50 Prozent der Werte begrenzen, und die Ausreißer in Box-Plots dargestellt. Die Entwicklung der Meinungsänderungen wurde zusätzlich in Häufigkeitsverteilungen beschrieben.

5.3 Qualitative Erhebung

Mit Hilfe der quantifizierten Ergebnisse des Leistungstests zu den informatischen Grundkonzepten kann das Unterrichtskonzept nur sehr begrenzt beurteilt werden, denn verschiedenste Einflussgrößen auf den Lernerfolg, förderliche oder behindernde Rahmenbedingungen etc. bleiben unberücksichtigt. Diese können nur mit Hilfe qualitativer Forschungsmethoden erfasst werden.

5.3.1 Forschungstagebuch

Im Laufe des Schuljahrs stellte sich heraus, dass die Arbeit mit der Klasse montags nach dem Wochenende sehr viel schwieriger als donnerstags war. Diese Art von Einflussgrößen auf den Unterrichtsversuch dokumentierte die Forscherin über das gesamte Schuljahr hinweg in einem Forschungstagebuch. Vor allem das kontinuierliche Beschreiben der Stimmungslage der Schülerinnen und Schüler während des Unterrichts und der in den einzelnen Unterrichtseinheiten aufgetretenen Besonderheiten wurden festgehalten.

Material

Zu Beginn des Schuljahrs wurde mit einem stark strukturierten Forschungstagebuch gearbeitet, das folgende Kategorien enthielt:

- Fach, Datum, Uhrzeit
- Thema des Fachunterrichts
- Informatische Grundkonzepte
- Explizit angesprochene informatische Grundkonzepte; Methoden und Hilfsmittel zu ihrer Vermittlung
- Eingesetzte Mikrowelten, verwendete Logo-Befehle
- Produkte, Präsentationen der Schülerinnen und Schüler
- Verteilung der Unterrichtszeit auf Lehrervortrag, Klassengespräch, Arbeit am Computer, Präsentation, Schülerarbeit ohne Computer

- Verteilung der Unterrichtszeit auf die Fachinhalte, Logo und das Anwenden der eigenen Produkte
- Stundenverlauf (verbal und als Rating-Skala)
- Motivation der Schülerinnen und Schüler (verbal und als Rating-Skala)
- Schwierigkeitsgrad der Fachinhalte, der informatischen Grundkonzepte und Logo (verbal und als Rating-Skala)
- Probleme mit der Programmiersprache, den informatischen Grundkonzepten und Fachinhalten
- Sonstige Schwierigkeiten.

Im Laufe der ersten Schulwochen stellte sich heraus, dass die Struktur des Forschungstagebuchs nicht flexibel genug war. Viele Kategorien wurden kaum verwendet. Andererseits war es teilweise schwierig, wichtige Ereignisse im Schema festzuhalten. Aus diesem Grund wurde die Struktur nach den Herbstferien folgendermaßen angepasst:

- Fach, Datum, Uhrzeit
- Thema des Fachunterrichts
- Informatische Grundkonzepte
- Eingesetzte Mikrowelten, verwendete Logo-Befehle
- Produkte, Präsentationen der Schülerinnen und Schüler
- Verteilung der Unterrichtszeit auf Lehrervortrag, Klassengespräch, Arbeit am Computer, Präsentation, Schülerarbeit ohne Computer.

Daran anschließend konnten dann in offener Form weitere Anmerkungen angefügt werden. Da vor allem die letzten Kategorien häufig nicht genutzt und die meisten Eintragungen im offenen Teil am Ende der Vorlage gemacht wurden, gab es Ende Februar eine weitere Änderung. Das Forschungstagebuch enthielt nur noch die organisatorischen Daten wie Fach, Datum und Zeit und gliederte den offenen Teil in die Abschnitte

- allgemeine Stimmung,
- Arbeit,
- Besonderheiten,
- Schwierigkeiten und
- Sonstiges.

Durchführung

Die Forscherin selbst schrieb das Forschungstagebuch in der Regel direkt im Anschluss an die Unterrichtsstunden.

Auswertung

Bei der Auswertung der einzelnen Unterrichtseinheiten (vgl. Kapitel 4 und 6.4) wurden die entsprechenden Aufzeichnungen des Forschungstagebuchs auf besondere Situationen im Unterricht hin evaluiert.

5.3.2 Leitfadeninterview Lehrerinnen

Zur qualitativen Befragung der Lehrerinnen bezüglich des gesamten Unterrichtsversuchs wurde ein Interviewleitfaden (vgl. Anhang E) entwickelt. Mit ihm wurden Informationen zum Unterrichtskonzept, seiner Wirkung auf die Schülerinnen und Schüler und die Integration in den Fachunterricht aus der Perspektive der beobachtenden und teilweise sehr berufserfahrenen Lehrerinnen gewonnen.

Material

Der Lehrerinnen-Interviewleitfaden umfasste zwölf Bereiche:

- | | |
|-----------------------------------|--|
| • Arbeitsblätter | • Lernerfolg |
| • Logo-Mikrowelten | • Motivation und Ausdauer der Schülerinnen und Schüler |
| • Unterrichtsformen | • Rolle der Forscherin |
| • Projektarbeit | • Rahmenbedingungen |
| • Hausaufgaben | • Fortführung des Konzepts durch die Fachlehrerinnen. |
| • Integration in die Schulfächer | |
| • Fächerübergreifender Unterricht | |

In der Auswertung wurden diese zu folgenden Hauptthemen zusammengefasst:

- Beurteilung des Gesamtkonzepts auf Basis der Einschätzung
 - der Eignung des Gesamtkonzepts für Klassenstufe 5
 - der Unterrichtsformen
 - des fächerübergreifenden Ansatzes
 - der Arbeitsmaterialien
 - der Projektarbeit
- Erfassung der Lernergebnisse und Einflussfaktoren
 - Fachinhalte
 - informatische Grundkonzepte und Computerbedienung
 - allgemeine Fähigkeiten
 - Förderung und/oder Benachteiligung bestimmter Gruppen von Schülerinnen und Schülern
 - Computer AG
- Verhalten der Schülerinnen und Schüler im Unterricht
 - Motivation
 - Ausdauer
 - Bedeutung des Unterrichts
 - Kooperation
- Einfluss der Rahmenbedingungen auf den Unterrichtsversuch
 - Aufteilung des Unterrichts auf zwei Lehrerinnen in einem Fach
 - Unterrichtsräume
 - Doppelstunden

- Partnerarbeit am Computer
- Hausaufgaben
- Einschränkung der Arbeit auf Logo-Systeme
- Benotung
- Unterrichtskonzeption und -durchführung durch die Forscherin
- weitere Rahmenbedingungen (Stundenverteilung, Belohnung, etc.).

5.3.3 Leitfadeninterview Schülerinnen und Schüler

Auch für die Schülerinnen und Schüler wurde ein Interviewleitfaden (vgl. Anhang F) entworfen. Mit ihm wurden Informationen über den Unterrichtsversuch, die erlernten Inhalte, die Selbsteinschätzung des Verhaltens im Unterricht und den Einfluss der Rahmenbedingungen gewonnen. Den Unterricht zur Vermittlung der informatischen Grundkonzepte nannten die Schülerinnen und Schüler Computerunterricht.

Material

Der Interviewleitfaden für die Schülerinnen und Schüler gab folgende Themenschwerpunkte vor:

- | | |
|--|---|
| • Einstellungen zum Computerunterricht | • Logo |
| • Mitarbeit | • Ritterprojekt |
| • Vergleich Computerunterricht – Fachstunden | • Lerninhalte |
| • Zusammenarbeit | • Fortführung des Computerunterrichts |
| • Arbeitsblätter | • Bedeutung des Computerunterrichts außerhalb der Schule. |

Für die Auswertung ergaben sich daraus die Hauptthemen

- Beurteilung des Gesamtkonzepts auf Basis der Einschätzung
 - der Stunden im Computer- und Fachunterricht
 - der Arbeitsmaterialien
 - der Projektarbeit
- Persönliche Bedeutung des Computerunterrichts
 - Wunsch nach Fortführung
 - Gesprächsthema bei den Eltern und Freunden
- Erfassung der Lernergebnisse bezüglich
 - der Fachinhalte in Mathematik, Englisch, Deutsch und Musik
 - der Computerbedienung
- Schülerverhalten im Unterricht
 - Mitarbeit
 - Partnerarbeit am Computer

- Einstellungen zur Arbeit im Computerunterricht
- gegenseitige Hilfe
- Logo
 - Vergleich zwischen der Arbeit mit Logo und anderen Programmen
 - negative Aspekte von Logo
 - Vergleich zwischen MSW Logo und Imagine Logo
 - Besonderheiten in Imagine Logo.

5.3.4 Unterrichtsmaterialien zu einzelnen Unterrichtseinheiten

Aussagen über das Erlernen einzelner informatischer Grundkonzepte können nicht allein aus Ergebnissen eines Leistungstests abgeleitet werden. Vielmehr ist es notwendig, die in alltäglichen Unterrichtssituationen entstandenen Materialien wie Arbeitsblätter, Tests und Programme zu evaluieren. Sie geben detailliert Auskunft, welche Teile des Unterrichtskonzepts erfolgreich waren und an welchen Stellen Veränderungen, Anpassungen oder Vertiefungen vorgenommen werden müssen.

Stichprobe

Der Stichprobenumfang bei den Arbeitsblättern variiert sehr stark. Eine Schülerin konnte über sechs Wochen von Mitte Mai bis Ende Juni nicht am Unterricht teilnehmen, so dass von ihr aus dieser Zeit keine Ergebnisse vorliegen. Abweichungen von der Stichprobengröße von 30 Kindern sind in Kapitel 6.4 bei der Darstellung der Auswertungsergebnisse der Unterrichtsmaterialien angegeben. An den beiden Tests zu den informatischen Grundkonzepten, die Ende November und Mitte Februar geschrieben wurden, nahmen alle Kinder teil. Da die Logo-Programme in Paaren erstellt wurden, ist der Stichprobenumfang bei diesen maximal 15. Weniger sind es zum Beispiel dann, wenn Arbeiten nicht abgespeichert wurden.

Material

Zur Überprüfung des Lernerfolgs einzelner informatischer Grundkonzepte wurden Unterrichtsmaterialien ausgewählt, die diese zum Inhalt hatten und die Rückschlüsse auf das Erlernen ermöglichten. Somit wurden die Arbeitsblätter ausgeschlossen, die im Klassenverband bearbeitet wurden.

Auswertung

Die Arbeitsblätter, Logo-Programme und Aufgaben der Tests wurden von der Forscherin in Abhängigkeit der durch die zu überprüfenden informatischen Grundkonzepte vorgegebenen spezifischen Fragestellungen ausgewertet.

6 Darstellung und Interpretation der Ergebnisse

Die Darstellung und Interpretation der Ergebnisse gliedert sich in fünf Abschnitte. Die Eignung des Gesamtkonzepts zur Vermittlung der informatischen Grundkonzepte in Mathematik, Englisch, Deutsch und Musik wird auf Basis der Ergebnisse der Interviewdaten beurteilt. Die Projektarbeit an der Rittergeschichte wird in einem eigenen Kapitel diskutiert. Darauf folgt die quantitative Auswertung der Tests zu den informatischen Grundkonzepten. Die Ergebnisse werden hinsichtlich ihrer Abhängigkeit vom Geschlecht und den unterschiedlichen Fähigkeiten der Kinder analysiert. Anschließend wird der Lernerfolg der Klasse bezüglich einzelner informatischer Grundkonzepte qualitativ beschrieben. Dies erfolgt auf Basis der Arbeitsblätter und Logo-Programme und der qualitativ ausgewerteten Teile der Tests. Der letzte Abschnitt beschäftigt sich schließlich mit der Veränderung der Motivationen und Einstellungen der Schülerinnen und Schüler im Umgang mit dem Computer.

Abbildung 6.1 gibt einen Überblick über die Aspekte der Datenauswertung und ihrer Wechselbeziehungen zur Beurteilung des Gesamtkonzepts.

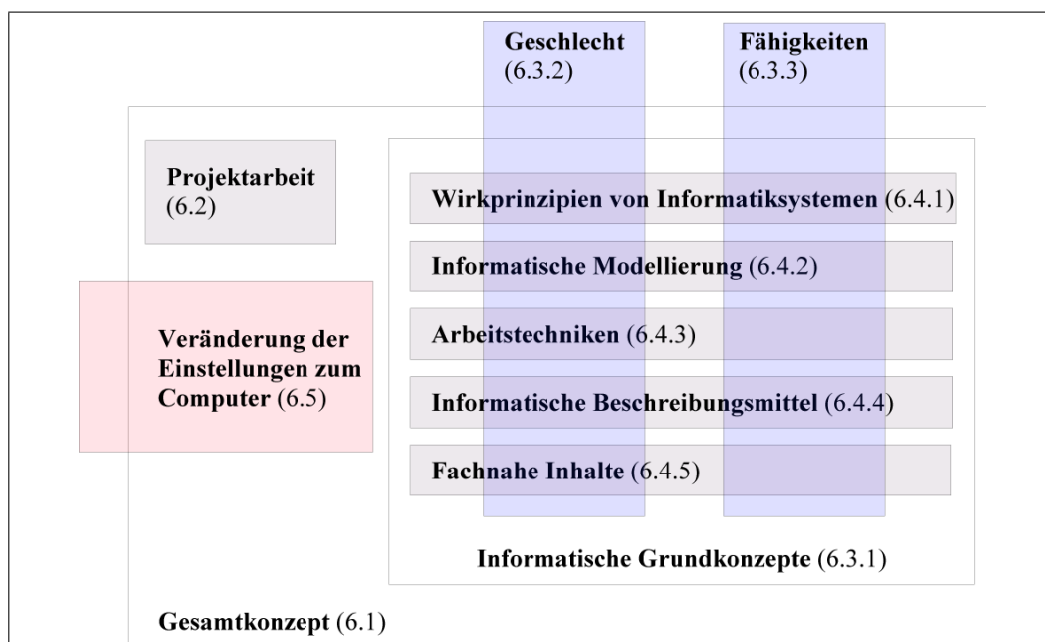


Abbildung 6.1: Überblick Evaluation

6.1 Gesamtkonzept

Die Durchführung des Unterrichtsversuchs hat gezeigt, dass Schülerinnen und Schülern bereits zu Beginn der Sekundarstufe I in einem kontinuierlichen Prozess informatische Grundkonzepte im Fachunterricht vermittelt werden können. Trotz der sehr hohen Anforderungen für die Klassenstufe 5 aufgrund

- des gleichzeitigen Lernens von Fachinhalten und informatischen Grundkonzepten,
- der Umsetzung von Problemlösungen in Logo und
- der häufig fachintegrativen und fächerübergreifenden Lern- und Arbeitsformen

ist das Konzept an keiner Stelle eingebrochen. Der in Kerres (2001, S. 97) dargestellte Neuigkeitseffekt durch den Einsatz der neuen Medien, der schnell abklingt und somit zu einem Nachlassen der Lernmotivation führt, war, anders als von Schrage (1994) und Clark (1985) in einigen Studien kritisiert, nicht ausschlaggebend für den Erfolg. Vielmehr wurde deutlich, dass permanenter Computereinsatz im Fachunterricht auch mit dem Ziel möglich ist, informatische Grundkonzepte zu vermitteln. Der Unterricht läuft dann wie in anderen Schulfächern ab.

Detaillierte Beschreibungen von Stärken und Schwächen des Unterrichtskonzepts werden aus dem Datenmaterial der Interviews der Lehrerinnen als Beobachtende und Schülerinnen und Schüler als Betroffene abgeleitet.

6.1.1 Verfahren zur Auswertung der Interviewdaten

Zur Auswertung der Interviewdaten der Lehrerinnen und der Schülerinnen und Schüler wurde die induktive Kategorienbildung nach Mayring eingesetzt. Es handelt sich dabei um eine Technik der zusammenfassenden qualitativen Inhaltsanalyse. Dies hat den Vorteil, dass die Kategorien direkt aus dem Material in einem Verallgemeinerungsprozess abgeleitet werden, ohne dass vorher ein Theoriekonzept entwickelt wird, auf das Bezug genommen werden muss (Mayring, 2003, S. 75). Aufgrund der geringen Anzahl von Interviews (drei Lehrerinnen und sechs Schülerinnen und Schüler) wurde auf den Schritt der Revision des Kategoriensystems und die damit verbundene Rückkopplung zur erneuten Bearbeitung des kompletten Ausgangsmaterials verzichtet. Daraus ergibt sich das in Abbildung 6.2 dargestellte Prozessmodell.

In der ersten Phase *Ziel der Analyse* wird die Fragestellung für die Auswertung der Interviews präzisiert. Bei der *Definition des Ausgangsmaterials* werden die Interviewdaten und ihre Entstehungssituation beschrieben. Das Material, das Ausgangspunkt der Kategoriendefinition ist, wird durch die *Festlegung des Selektionskriteriums* in Abhängigkeit der zuvor definierten Fragestellung bestimmt. Zur Zusammenfassung der Inhalte muss ein *Abstraktionsniveau* festgelegt werden. Bei der *lokalen Materialdurcharbeitung* und *Kategorienformulierung* wird das Material Zeile für Zeile untersucht. Entspricht eine

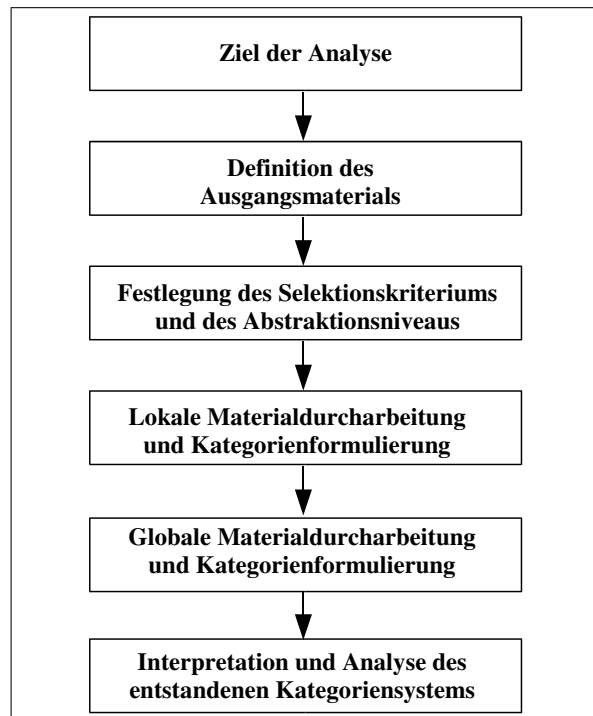


Abbildung 6.2: Prozessmodell der induktiven Kategorienbildung

Aussage einem Selektionskriterium, so wird unter Berücksichtigung des Abstraktionsniveaus die erste Kategorie formuliert. Weiter selektierte Textstellen werden entweder in bereits vorhandene Kategorien einsortiert oder sie bilden selbst eine neue Kategorie (Mayring, 2003, S. 76). In dieser Arbeit wurden während der lokalen Materialdurcharbeitung nur Kategorien innerhalb eines Themenkomplexes gebildet. Bei der *globalen Materialdurcharbeitung und Kategorienformulierung* werden die zusammengehörenden Kategorien in Analogie zur Revision bei Mayring (2003, S. 75) identifiziert, zusammengeführt und entsprechend erweitert. Allerdings wird das gesamte Ausgangsmaterial nicht noch einmal durchgearbeitet. Zum Abschluss wird das entstandene Kategoriensystem interpretiert und analysiert.

Die einzelnen Auswertungsschritte werden im Folgenden erst für die Auswertung der Interviews der Lehrerinnen und dann der Schülerinnen und Schüler konkretisiert.

Lehrerinnen-Interviews

Ziel der Analyse: Aus den Interviewdaten der Lehrerinnen sollten induktiv Aussagen über

- die Qualität des Gesamtkonzepts (Arbeitsmaterial, Projekt, Unterrichtsformen, fächerübergreifender Ansatz),

- die Lernergebnisse und Einflussfaktoren,
- das Schülerverhalten im Unterricht und
- den Einfluss der gegebenen Rahmenbedingungen auf den Unterrichtsversuch

abgeleitet werden.

Definition des Ausgangsmaterials: Das Material bestand aus drei Interviews, die die Forscherin mit den Lehrerinnen am Ende des Unterrichtsversuchs durchführte. Grundlage war der Interviewleitfaden aus Anhang E. Die Lehrerinnen konnten aufgrund ihrer Teilnahme an den Fortbildungen (vgl. Kapitel 3.5) und ihren Beobachtungen im Unterricht die Fragen qualifiziert beantworten.

Die Transkription der Tonbandaufzeichnungen erfolgte durch die Forscherin selbst. Es wurden lediglich die Interviewtexte transkribiert, parasprachliche Elemente wie Pausen oder Stimmlagen jedoch nicht berücksichtigt.

Festlegung des Selektionskriteriums und des Abstraktionsniveaus: Für die lokale Materialdurcharbeitung der zwölf Bereiche (vgl. Kapitel 5.3.2, Anhang E) wurden folgende Abstraktionsniveaus definiert:

- *Arbeitsblätter:* Alle Bemerkungen, die Form und Inhalt bewerten
- *Logo-Mikrowelten:* Aussagen bezüglich der Eignung der Mikrowelten für die 5. Klassenstufe und ihre Erweiterungsmöglichkeiten
- *Unterrichtsformen:* Bewertung der Aufteilung von Frontal- und Arbeitsphasen am Computer und des Arbeitens an theoretischen Konzepten mit und ohne Computer
- *Projektarbeit:* Alle konkreten Charakterisierungen bezüglich des Projektthemas, des Verhaltens der Schülerinnen und Schüler im Projekt, der Projektrahmenbedingungen und der Projektergebnisse
- *Hausaufgaben:* Einschätzungen bezüglich ihrer Notwendigkeit zur Vermittlung von informatischen Grundkonzepten
- *Integration in die Schulfächer:* Auswirkungen der Vermittlung der informatischen Grundkonzepte auf den Fachunterricht und die Eignung der Verknüpfung der informatischen Grundkonzepte mit konkreten Fachinhalten
- *Fächerübergreifender Unterricht:* Aussagen zur fächerübergreifenden Vermittlung der informatischen Grundkonzepte und der Wiederholung dieser in verschiedenen Fachkontexten
- *Lernerfolg:* Beschreibung der erlernten Inhalte, der Vor- und Nachteile für einzelne Schülerinnen und Schüler, der Einflussfaktoren auf das Durchhaltevermögen der Klasse und der Auswirkungen des Besuchs der Computer AG
- *Motivation und Ausdauer:* Einschätzung der Motivation und Ausdauer der Schülerinnen und Schüler bei der Vermittlung der informatischen Grundkonzepte und Angabe von beeinflussenden Faktoren
- *Rolle der Forscherin:* Bewertung der Forscherin als Lehrerin
- *Rahmenbedingungen:* Konkrete Aussagen über die Auswirkungen der Rahmenbe-

dingungen (Stundenverteilung, Partnerarbeit am Computer, Aufteilung des Unterrichts in den beteiligten Fächern auf zwei Lehrerinnen, fehlende Präsentationsmöglichkeiten im Klassenzimmer, Einschränkung auf die Arbeit mit Logo-Systemen etc.) auf den Erfolg des Gesamtkonzepts

- *Fortführung des Konzepts*: Begründung, warum die Vermittlung der informatischen Grundkonzepte mit Logo-Systemen von den Lehrerinnen nicht fortgesetzt wird und Einschätzung, ob das Konzept überhaupt für die Klassenstufe 5 geeignet ist.

Lokale Materialdurcharbeitung und Kategorienformulierung: Die Forscherin wertete die zwölf Themenbereiche zunächst lokal aus und formulierte Kategorien.

Globale Materialdurcharbeitung und Kategorienformulierung: Das gesamte Datenmaterial der Interviews wurde ein zweites Mal durchgearbeitet, um das Kategoriensystem auf Basis von relevanten Aussagen der verschiedenen Themenbereiche zu verdichten.

Interpretation und Analyse des entstandenen Kategoriensystems: Das entwickelte Kategoriensystem wurde bezüglich der vier Ziele der Analyse zusammengestellt und übergreifend interpretiert.

Schülerinnen- und Schüler-Interviews

Bezüglich der Auswertung der Interviews der Schülerinnen und Schüler werden nur die Auswertungsschritte charakterisiert, die sich von denen der Lehrerinnen-Interviews unterscheiden.

Ziel der Analyse: Zielsetzung der Interviews war die Erhebung von Daten über das Unterrichtskonzept aus Sicht der Betroffenen. Induktiv sollten dabei Aussagen über

- das Gesamtkonzept (integrierte Unterrichtsform, Arbeitsmaterial, Projektarbeit),
- die persönliche Bedeutung des Unterrichts für die Schülerinnen und Schüler,
- das Lernen sowohl der Computerbedienung unter Berücksichtigung der informatischen Grundkonzepte als auch der Fachinhalte,
- das Schülerverhalten im Unterricht und
- die Eignung von Logo als Denk- und Arbeitsmittel

gewonnen werden.

Definition des Ausgangsmaterials: Das Material bestand aus Interviewdaten mit drei Schülerinnen und drei Schülern, die jeweils starke, mittlere und schwache Ergebnisse in den Tests zu den informatischen Grundkonzepten im Laufe des Schuljahrs (vgl. Anhang G) und dem Nachtest zu den informatischen Grundkonzepten am Ende des Unterrichtsversuchs erzielt hatten. Grundlage der Interviews war der Interviewleitfaden in Anhang F. Die Transkription der Tonbandaufzeichnungen der Interviews erfolgte analog zu den Lehrerinnen-Interviews.

Festlegung des Selektionskriteriums und des Abstraktionsniveaus: Auch der Interviewleitfaden für die Schülerinnen und Schüler erforderte durch seinen Aufbau (vgl. Kapitel 5.3.3 und Anhang F) die Definition von zehn verschiedenen Abstraktionsniveaus für die Materialdurcharbeitung.

- *Einstellungen zum Computerunterricht:* Aussagen bezüglich der Wichtigkeit des Computerunterrichts und der dort erstellten Arbeiten
- *Mitarbeit:* Eigene Einschätzung und Begründung
- *Vergleich Computerunterricht – Fachunterricht:* Einschätzung des Schwierigkeitsgrads und Aussagen über Unterschiede
- *Zusammenarbeit:* Charakterisierung der Partnerarbeit an den Computern, der gegenseitigen Hilfestellungen und der Beobachtung der Arbeiten anderer Partner
- *Arbeitsblätter:* Anmerkungen über das Lesen und Verstehen
- *Logo:* Vergleichende Aussagen zwischen Imagine Logo und MSW Logo, Logo im Allgemeinen und dem Arbeiten mit anderen Computerprogrammen und Beschreibung der Besonderheiten von Imagine Logo
- *Ritterprojekt:* Charakterisierung der Veränderungen im Unterricht während der Arbeit an der Rittergeschichte, empfundene Auswirkungen und Begründung der geringen Experimentierfreudigkeit bei vorgegebenen Aufgabenstellungen
- *Lerninhalte:* Beschreibung der erlernten informatischen und fachlichen Inhalte
- *Fortführung des Computerunterrichts:* Aussagen über den Wunsch, den Computerunterricht fortzuführen und konkrete Veränderungsvorschläge
- *Bedeutung des Computerunterrichts außerhalb der Schule:* Darstellungen über die Thematisierung.

Die lokale und globale Materialdurcharbeitung und Kategorienformulierung und die Interpretation der Kategorien erfolgte entsprechend dem Vorgehen bei den Lehrerinnen.

Die folgenden Unterkapitel nehmen nur auf die Interviewdaten Bezug, die für die Beurteilung des Gesamtkonzepts und speziell des Projektunterrichts relevant waren. Die Aussagen der Lehrerinnen zum Bereich *Rolle der Forscherin* lieferten zum Beispiel keine wesentlichen Informationen und wurden aus diesem Grund vernachlässigt.

6.1.2 Aspekte des Gesamtkonzepts

Eignung für Klassenstufe 5

Aus Sicht aller Lehrerinnen ist die Klassenstufe 5 prinzipiell zur Vermittlung der informatischen Grundkonzepte geeignet. Ihrer Meinung nach sind die informatischen Grundkonzepte für die Schülerinnen und Schüler nicht zu schwierig. Sie argumentierten, dass diejenigen, die sie in Klasse 5 nicht verstehen, sie auch zu einem späteren Zeitpunkt nicht lernen würden. Die Mathematiklehrerin, die eine Überforderung bei wenigen Kindern sah, bezog diese auf die sehr hohen kognitiven Leistungen des geistigen Vorweg-

nehmens von Abläufen beim Programmieren. Sie wies außerdem auf die Schwierigkeit hin, dass die Kinder in Klasse 5 mit sehr viel Neuem konfrontiert würden. Aus diesem Grund nannte sie Klasse 6 und 7 als Alternative für einen Einstieg in die informatischen Grundkonzepte.

Vergleich Computer- und Fachunterricht

Auch die Schülerinnen und Schüler schätzten die Anforderungen im Unterricht zu den informatischen Grundkonzepten als nicht zu hoch ein (vgl. Tabelle 6.1). Drei von ihnen beurteilten den Unterricht sogar als einfacher als den Fachunterricht. Nur ein Schüler fand ihn schwieriger. Dies stand im Widerspruch zu den Aussagen von fünf Schülerinnen und Schülern, dass sie die Inhalte nicht sofort verstanden hätten. Unterschiede im

III, 1) Fandest du den Computerunterricht schwerer als den normalen Unterricht in den Fächern Mathematik, Deutsch, Englisch und Musik?	Anzahl
schwerer	1
leichter	3
abhängig vom Fach	1
widersprüchliche Aussage	1

Tabelle 6.1: Schwierigkeitsgrad des Computer- und Fachunterrichts im Vergleich (6 Schülerinnen und Schüler)

Computer- und Fachunterricht stellten die Kinder vor allem im Bereich der Rahmenbedingungen (Räume, Bücher, Hausaufgaben, Klassenarbeiten), der Arbeitsformen (Arbeit am Computer, Bearbeitung von Arbeitsblättern) und beim Schreiben fest. Diese deuten jedoch auf keinerlei Überforderung hin. Auch die von den Kindern genannten fachspezifischen Unterschiede (vgl. Tabelle 6.2) bestätigen diese Annahme. Allein ein Schüler hatte sprachliche Schwierigkeiten bei der Bedienung des englischen Imagine Logo Systems.

III, 2) Was war im Computerunterricht eigentlich anders als im normalen Unterricht?	Anzahl
kein richtiges Englisch	1
in Imagine Logo nur englische Kommunikation möglich; die Englischlehrerin spricht auch mal Deutsch	1
weniger Schreiben	1
keine Arbeit mit Zahlen	1
Umgang mit dem Computer ist für später wichtig, Englisch nicht	1
keine Angabe	1

Tabelle 6.2: Computer- und Fachunterricht im Vergleich (6 Schülerinnen und Schüler)

Das Arbeitstempo beim Lösen der Aufgaben beurteilten die Schülerinnen und Schüler als angemessen. Insbesondere zur Förderung des explorativen Lernens mit Logo stand

ihnen genügend Zeit zur Verfügung, so dass ein Experimentieren möglich war. Zwei von ihnen hätten teilweise auch schneller arbeiten können.

Unterrichtsformen

Frontalphasen zur Einführung theoretischer Konzepte waren nach Ansicht der Lehrerinnen notwendig. Sie müssten durch eine bessere Mischung mit Arbeitsphasen am Computer noch verkürzt werden. Insgesamt äußerten alle Lehrerinnen den Wunsch, den Anteil der praktischen Arbeit zu erhöhen, um den Schülerinnen und Schülern noch mehr Zeit zum Ausprobieren und Experimentieren zur Verfügung zu stellen. Im Zusammenhang

III, 1) Der Computerunterricht war häufig in Frontalphasen und in Gruppenarbeit am Computer aufgeteilt. Wie beurteilen Sie diese Aufteilung?	Anzahl
bessere Mischung, mehr Methodenwechsel	1
mehr praktische Arbeit am Computer	3
kürzere Frontalphasen	1
III, 2) Der Computerunterricht fand für die Schülerinnen und Schüler nicht nur am Computer statt. Wie beurteilen Sie die Aufteilung zwischen Computerarbeitsphasen und Arbeitsphasen ohne Computer?	Anzahl
weniger Theorie	1
kürzere Theoriephasen	1
methodische Kniffe in Theoriephasen stärker berücksichtigen	2

Tabelle 6.3: Frontalphasen, Partnerarbeit am Computer und Theorieunterricht
(3 Lehrerinnen, Mehrfachnennungen möglich)

damit standen die Aussagen zum Theorieunterricht. Nach Meinung der Deutschlehrerin sollte er insgesamt eingeschränkt werden. Die Mathematiklehrerin forderte zumindest eine Verkürzung der zusammenhängenden Theoriephasen und zwei der Lehrerinnen wünschten sich eine stärkere Berücksichtigung methodischer Tricks. Die Trennung von theoretischen und praktischen Arbeiten bzw. Frontalunterricht und Partnerarbeit am Computer wurde nach Aussage der Mathematiklehrerin durch die Arbeit in zwei verschiedenen Unterrichtsräumen noch verstärkt.

“Dann war das immer so für viele Schüler einfach so, im Klassenzimmer, da ist halt das Übel, da müssen wir jetzt wieder Theorie machen und dann gehen wir in den Computerraum und da wird’s dann interessant. Dort dürfen wir was Schönes machen.“ (LM, S. 6, 37 ff.¹)

Es ist also sinnvoll, dass der gesamte Unterricht im Computerraum stattfindet, damit den Schülerinnen und Schülern die Grenzen und Übergänge zwischen den theoretischen und praktischen Teilen der Arbeit nicht so deutlich werden. Dies erfordert, dass die

¹Erläuterung zur Codierung: Die drei Lehrerinnen wurden mit LM (Mathematik), LE (Englisch) und LD (Deutsch und Musik) abgekürzt. Die Codierung besteht aus der Bezeichnung der Lehrerin, gefolgt von der Seitenzahl und den Zeilennummern im Transkript des entsprechenden Interviews.

schwierigen und häufig zeitintensiven Wechsel zwischen Frontalphasen und Arbeiten am Rechner mit Hilfe von Ritualen konsequent eingeübt werden.

Fächerübergreifender Ansatz

Das Design des Unterrichtskonzepts zur Vermittlung der informatischen Grundkonzepte in Mathematik, Englisch, Deutsch und Musik war fächerübergreifend konzipiert. Wie aus Tabelle 6.4 hervorgeht, fand der Unterricht aus der Perspektive der Lehrerinnen nicht immer fächerübergreifend statt. Interessanterweise wurde von der Deutschlehrerin der Unterricht zu den informatischen Grundkonzepten als Anker für eine Abstimmung der Fächer angegeben.

„Also ich denk, das war eigentlich auch so bisschen der Aufhänger, dass wir da, ja, dass wir einfach uns abgestimmt haben, damit man das dann auch im Computer dann machen kann.“ (LD, S. 10, 30 f.)

VII, 1) Wurden die Inhalte fächerübergreifend vermittelt oder liefen die Stunden in den einzelnen Fächern nebeneinander her?	Anzahl
zu Beginn des Schuljahrs nicht fächerübergreifend	1
teilweise nicht fächerübergreifend	1
fächerübergreifend	1
Aufhänger für die Abstimmung zwischen den Fächern	2
VII, 1a) Können Sie Beispiele angeben?	Anzahl
s-Genitiv	1
Grammatik	1
Bruchrechnen, Noten	2
Ritterprojekt	2
Befehle	1
Maschinendarstellung von Funktionen	1

Tabelle 6.4: Fächerübergreifender Unterricht (3 Lehrerinnen, Mehrfachnennungen möglich)

Die Befürchtung, die Kinder durch mehrfaches Wiederholen der informatischen Inhalte in den verschiedenen Fachkontexten zu langweilen, wurde von den Lehrerinnen nicht bestätigt.

Arbeitsmaterialien

Da im Unterricht ohne Buch gearbeitet wurde, erhielten die Schülerinnen und Schüler sehr viele Arbeitsblätter. Ihren klaren Aufbau bewerteten alle Lehrerinnen sehr positiv. Allerdings waren die Blätter ihrer Meinung nach zu lang, enthielten zu viel Text und waren manchmal in einer für Kinder schwierig zu verstehenden Sprache formuliert. Dies bestätigten vor allem die beiden schwächeren Schülerinnen und Schüler, die die Arbeitsblätter nicht gerne gelesen haben und eigentlich schneller am Computer hätten arbeiten wollen. Bei den anderen war dies nicht der Fall. Sie schätzten die Arbeitsblätter

als Nachschlagewerke, um Inhalte selbstständig nachzulesen. Die beiden stärksten Kinder gaben außerdem an, dass die Arbeitsblätter für sie verständlich gewesen seien. Die Arbeitsblätter lassen sich in vier Gruppen einteilen:

- Bearbeitung im Klassenverband
- selbstständiges Durcharbeiten
- selbstständiges Durcharbeiten mit Angabe von Punkten
- Erarbeitung neuer Inhalte und gleichzeitiger Test.

Die Punkte, die den Schwierigkeitsgrad der Aufgaben beschrieben und einen Anreiz für die Schülerinnen und Schüler darstellen sollten, wurden von zwei Lehrerinnen als überflüssig angesehen. Sie waren der Meinung, dass die Kinder die Punkte überhaupt nicht wahrgenommen hätten (siehe Tabelle 6.5). Die Mathematiklehrerin bezeichnete

I, 2) Es wurden vier verschiedene Formen von Arbeitsblättern verwendet. Welche Formen haben sich bewährt?	Anzahl
Mischen verschiedener Bearbeitungsformen	2
Punkte nicht notwendig	2
selbstständiges Durcharbeiten	1
Erarbeitung neuer Inhalte und gleichzeitiger Test	1

Tabelle 6.5: Verschiedene Bearbeitungsformen der Arbeitsblätter (3 Lehrerinnen, Mehrfachnennungen möglich)

das eigenständige Erarbeiten der Inhalte mit integriertem Test als optimale Lösung. Ziel kann es ihrer Meinung nach zwar nicht sein, die Kinder durch Notendruck zur Arbeit anzuspornen, gleichzeitig sah sie es aber als eine sehr wirkungsvolle Methode an. Die Englischlehrerin präferierte das selbstständige Durcharbeiten der Arbeitsblätter. Da keine Bearbeitungsform besonders hervorgehoben wurde, scheint das von zwei Lehrerinnen vorgeschlagene Mischen verschiedener Formen am sinnvollsten.

Die eingesetzten Mikrowelten beschrieben die Lehrerinnen als sehr passend, offen, interessant und motivierend. Sie waren ihrer Ansicht nach aber nicht mit kommerziellen Programmen vergleichbar. Die Mathematik- und Deutschlehrerin äußerten sich auch über den ausschließlichen Einsatz von Logo-Systemen. Die Einschränkung vermied ihrer Meinung nach eine Überforderung der Schülerinnen und Schüler. Erstere war davon überzeugt, dass sich die Kinder nach einem Jahr Unterricht zu den informatischen Grundkonzepten leicht in gängige Programme einarbeiten werden können. Die andere widersprach sich dahingehend, dass sie es auch gut gefunden hätte, mit weiteren Programmen zum Versenden von E-Mails, zur Erzeugung von Notentabellen und Schreiben von Texten zu arbeiten.

Erfolg des Konzepts

Ein Indiz für den Erfolg des Gesamtkonzepts ist die Tatsache, dass die Mehrheit der

interviewten Schülerinnen und Schüler weiterhin gerne am Unterricht zu den informatischen Grundkonzepten teilnehmen würde. Diese Kinder bedauerten das Ende sogar, da sie im Computerunterricht mehr Spaß als im Fachunterricht hatten. Lediglich die beiden Schwächeren waren froh, dass der Computerunterricht zu Ende war. Das Mädchen wird allerdings die Projektarbeit vermissen.

Aus den in Tabelle 6.6 genannten Änderungswünschen können keine grundsätzlichen Veränderungen des Konzepts abgeleitet werden. Drei Schülerinnen und Schüler betonten, dass ihnen der Unterricht gut gefallen habe. Ein Kind wünschte sich eine Reduzierung des Stundenumfangs. Ein weiterer Schüler wollte nicht mehr an einem Ritterprojekt arbeiten. Bei dieser Aussage ging aus den Interviewdaten nicht hervor, ob damit das Thema oder die Projektarbeit an sich angesprochen wurde. Da die Projektarbeit von den anderen Mitschülerinnen und Mitschülern und den Lehrerinnen ansonsten sehr positiv bewertet wurde und auch die Arbeitsergebnisse der Kinder teilweise herausragend waren, wird die Projektarbeit ein wesentlicher Bestandteil des Gesamtkonzepts bleiben. Ein weiterer Wunsch bestand darin, länger an einer Sache arbeiten zu dürfen. Dies widersprach den Aussagen bezüglich des Arbeitstempos im Unterricht. Eine weitere Änderung betraf neue Spiele. Hier dachte der Schüler sicherlich an die Ausweitung der Programmierung von Spielen. Dass nur ganz wenige Lieder im Laufe des Schuljahrs gesungen wurden, lag an der in Kapitel 3.2 dargestellten Besonderheit der Projektstunde in Klassenstufe 5, so dass die Klasse keinen regulären Musikunterricht hatte.

IX, 2a) Stell dir vor, ich mache im nächsten Schuljahr in deiner Klasse den Computerunterricht so wie in diesem Schuljahr weiter. Was sollte dann anders sein?	Anzahl
weniger Unterrichtsstunden in der Woche	1
kein Ritterprojekt	1
länger an einer Sache arbeiten und trotzdem Abwechslung	1
neue Spiele	1
mehr Lieder singen	1
nichts	1

Tabelle 6.6: Änderungswünsche für den Computerunterricht (6 Schülerinnen und Schüler)

Zusammenfassung

Nach Ansicht der Lehrerinnen, Schülerinnen und Schüler ist das Gesamtkonzept für die Klassenstufe 5 geeignet. Der Unterricht wird von den Kindern nicht schwieriger als der Fachunterricht wahrgenommen und stellt keine zu hohen Anforderungen. Die Theoriephasen und der Frontalunterricht müssen gekürzt und methodisch abwechslungsreicher gestaltet werden. Fächerübergreifende Aspekte sollten vor allem zu Beginn des Schuljahrs in den kürzeren Unterrichtseinheiten stärker herausgearbeitet werden. Die für den

Unterricht konzipierten Arbeitsblätter bewerten die Lehrerinnen positiv. Sie weisen auf ihren klaren Aufbau hin und heben die Mischung der verschiedenen Bearbeitungsformen hervor. Teilweise müssen noch Formulierungen angepasst und kleinere Einheiten gebildet werden. Obwohl die Mikrowelten nicht mit kommerziellen Programmen vergleichbar sind, sind sie für den Unterricht geeignet. Der Erfolg des Gesamtkonzepts kann auf Basis der nur unwesentlichen Änderungswünsche der Schülerinnen und Schüler und der im folgenden Kapitel dargestellten Lernerfolge begründet werden.

6.1.3 Lernerfolge allgemein

Fachinhalte

Trotz der verkürzten Unterrichtszeit für die Vermittlung der reinen Fachinhalte durch die zusätzlichen informatischen Grundkonzepte konnten nach Angaben der Lehrerinnen alle Fachinhalte vermittelt werden. Allerdings musste gegen Ende etwas schneller gearbeitet und die Übungszeit insgesamt reduziert werden. Dies kann nach Aussage der Englischlehrerin in Jahrgangsstufe 6 durch Wiederholungen ausgeglichen werden. Sie hob auch eine positive Rückwirkung des Unterrichts zu den informatischen Grundkonzepten auf ihr eigenes Fach hervor. Sie nannte zusätzliche Übungen, die Vertiefung der Kenntnisse und eine Verdeutlichung der grammatikalischen Strukturen. Geeignete Verbindungen zwischen den informatischen Grundkonzepten und Fachinhalten sahen die Lehrerinnen in der Grammatik in Deutsch und Englisch und beim Thema Spiegelung und Parallelogramme in Mathematik. Hier lassen sich ihrer Meinung nach die informatischen Grundkonzepte optimal in das Fach integrieren.

Auch die Schülerinnen und Schüler erkannten, dass bei der Arbeit mit dem Computer Inhalte aus den Fächern erlernt wurden (vgl. Tabelle 6.7). Interessant ist, dass sie sowohl im Fachbereich Mathematik als auch Deutsch Programmablaufpläne nannten. Hier nahmen sie anscheinend nicht wahr, dass es sich bei Programmablaufplänen um allgemeine informatische Beschreibungsmittel und nicht um Inhalte des Fachs handelte. Gefragt nach den Inhalten, die ihnen im Computerunterricht besonders wichtig waren, hoben drei Kinder Themen aus dem Bereich Musik hervor. Dies bestätigte, dass paralleler Fachunterricht notwendig ist. Weiter nannte eine Schülerin das Thema Potenzen aus Mathematik. Für einen Schüler war eine Unterrichtsstunde wichtig, in der Dateien und Verzeichnisse behandelt wurden. Sie beinhaltet zwar die informatischen Grundkonzepte *Hierarchien* und *rekursive Datenstrukturen*, der Schüler sah jedoch die praxisorientierte Anwendung, die ihm beim Arbeiten mit dem Computer helfen wird.

Informatische Grundkonzepte und Computerbedienung

Die Schülerinnen und Schüler konnten nicht nach informatischen Grundkonzepten gefragt werden, die sie im Schuljahr erlernt hatten, da diese zum großen Teil nicht begrifflich eingeführt worden waren. Aus diesem Grund mussten sie über ihre Lernerfolge bezüglich der Bedienung des Computers berichten (siehe Tabelle 6.8).

Mathematik	Anzahl
Römische Zahlen	2
Potenzen	3
Schweizer Franken	1
Fadenbilder	1
Programmablaufpläne	1
Musik	Anzahl
Noten	5
Takt, Taktart	1
Rhythmus	1
Melodie	1

Deutsch	Anzahl
Satzbildung	2
Schreiben	3
Rechtschreibung	2
Geheimcode	1
Programmablaufpläne	2
Englisch	Anzahl
Wörter	5
nichts	1

Tabelle 6.7: Lernerfolge bezüglich der Fachinhalte im Computerunterricht (6 Schülerinnen und Schüler, Mehrfachnennungen möglich)

VIII, 1) Was hast du im Computerunterricht über den Computer und seine Bedienung gelernt? Erzähl mir ein paar Beispiele.	Anzahl
MSW Logo	3
Imagine Logo	2
Grundtechniken (öffnen, schreiben, speichern von Programmen)	3
Gestaltung	2
Programmierung	1
Fähigkeiten des Computers einschätzen	1
Bedienung eines Laptops	2
mit dem Computer umgehen	1

Tabelle 6.8: Lernerfolge bezüglich des Computers und seiner Bedienung (6 Schülerinnen und Schüler, Mehrfachnennungen möglich)

Die schwächeren Schülerinnen und Schüler nannten sehr konkrete Dinge wie MSW Logo und Imagine Logo. Im Gegensatz dazu gaben die leistungstärkeren Kinder allgemeine Grundtechniken wie das Öffnen, Schreiben und Speichern von Programmen an. Auch der kreative Umgang mit dem Computer wurde durch die Gestaltung hervorgehoben. Informatische Inhalte wurden zweimal angesprochen. Ein Kind erwähnte die Programmierung und ein anderes glaubte, durch den Unterricht die Fähigkeiten des Computers besser einschätzen zu können. Werden unter Programmierung Inhalte wie *Algorithmen*, *Befehle*, *Funktionen*, *Abstraktion*, *Modularisierung* etc. subsumiert, so haben beide die intendierten Lernziele des Unterrichts erreicht.

Allgemeine Fähigkeiten

Zentral für die Auswertung des Unterrichtskonzepts waren die Aussagen der Lehrerinnen über die Lernergebnisse. Sie nannten Fähigkeiten wie *Vorausdenken*, *Strukturieren*, *An-*

wenden von Schemata, genaues Arbeiten, Strategien zur Bewältigung von Frustrationen und Verstehen von Fehlermeldungen. Der Erfolg des Unterrichtskonzepts wird aufgrund dieser Feststellungen bestätigt. Die Kompetenzen sind nicht allein für die Arbeit mit dem Computer wichtig, sondern in vielen Bereichen hilfreich. Die Fähigkeit zu strukturieren unterstützt das Erschließen und Darstellen von Zusammenhängen eines Inhaltsbereichs. Informationen können leichter organisiert und geordnet werden. Dies ist sowohl für das eigene Lernen als auch zur Aufbereitung von Informationen für andere sinnvoll. Die immer stärkere Vernetzung der Welt und die damit wachsende Entscheidungskomplexität lässt die Bedeutung des vorausschauenden Denkens zunehmen. Häufig müssen unterschiedliche Vorgehensweisen analysiert, die sich aus ihnen ergebenden Konsequenzen abgewogen und auf Basis der Ergebnisse Entscheidungen getroffen werden. Auf der anderen Seite kann das Anwenden von Schemata von Routinetätigkeiten entlasten. Genaues Arbeiten spielt in allen Bereichen des Lebens eine Rolle. In der Schule und im Berufsleben müssen Aufgabenstellungen richtig erfasst und präzise gelöst werden. Gleiches gilt für die Frustrationstoleranz. Permanent werden Schülerinnen und Schüler mit Herausforderungen konfrontiert, die es zu bewältigen gilt. In der Schule werden sie dabei noch sehr stark von den Lehrerinnen und Lehrern betreut und motiviert. Im Berufsleben sind sie auf sich selbst gestellt. Wichtig sind dann Strategien, mit denen der sich bei Schwierigkeiten entwickelnde Frust abgebaut werden kann. Allein das genannte Verständnis von Fehlermeldungen bezieht sich auf den Umgang mit dem Computer. Bei den ersten fünf Punkten handelt es sich um Schlüsselqualifikationen, die den in den *Bildungsstandards für Informationstechnische Grundbildung* beschriebenen allgemeinen Fähigkeiten (vgl. Kapitel 2.2.2) zugeordnet werden können. Im hier vorgestellten Unterrichtskonzept bleibt es allerdings im Gegensatz zu den Bildungsstandards nicht offen, unter welchen konkreten Bedingungen diese Schlüsselqualifikationen gefördert werden. Entscheidend ist eine kontinuierliche, über einen längeren Zeitraum hin stattfindende Vermittlung der informatischen Grundkonzepte in systematischen Lernprozessen. Zusätzlich muss noch folgender Aussage der Englischlehrerin Beachtung geschenkt werden:

„Ja und sie haben wohl einen Grundstock, der ihnen hoffentlich bis ans Ende ihrer Schulzeit hier reicht.“ (LE, S. 8, 6 f.)

Da die *Informationstechnische Grundbildung* an Realschulen häufig als Anwendungs- und Produktschulung durchgeführt wird, nimmt die Lehrerin an, dass die informatischen Grundkonzepte eine Basis für das Kennenlernen von verschiedenen Programmen sei. Dies unterstützt die These, dass auch in einem benutzerorientierten Unterricht grundlegende und allgemeine informatische Konzepte vermittelt werden müssen.

Förderung und Benachteiligung

Sehr unterschiedlich waren die Aussagen der Lehrerinnen, welche Kinder von der Unterrichtskonzeption profitierten und welche durch sie benachteiligt wurden. Die Mathematiklehrerin schätzte die Situation für alle als vorteilhaft ein. Die anderen beschränkten es auf gute Schülerinnen und Schüler, auf Kinder mit großen Vorkenntnissen oder

diejenigen, die gewissenhaft veranlagt waren. Auf der anderen Seite reichte das Spektrum der Benachteiligten von niemandem, über Kinder mit Konzentrationsschwächen oder schlechtem abstrakten Denkvermögen bis zu Schülerinnen und Schülern, die nicht genau arbeiten konnten. Auch sie haben bis an das Schuljahresende durchgehalten. Ausschlaggebend dafür waren nach Meinung der Lehrerinnen verschiedene Gründe. Teilweise waren die Eltern sehr daran interessiert, dass ihre Kinder den Umgang mit dem Computer erlernten, so dass sie sie von der Wichtigkeit des Unterrichts überzeugten. Andere Schülerinnen und Schüler waren aufgrund ihres eigenen Interesses an den Inhalten sehr motiviert. Nach Ansicht der Deutschlehrerin war es manchmal auch die Fachlehrerin, die im Unterricht mit anwesend war, da sie die Autoritätsperson war, die die Kinder beurteilte. Als weiterer Grund für das Durchhaltevermögen wurde von der Englisch- und Deutschlehrerin die Computer AG genannt, da die Kinder dort freier an eigenen Themen arbeiten konnten. Eine versprochene Belohnung am Ende des Schuljahrs für die Kinder, die ihre Leistungen bezüglich der informatischen Grundkonzepte besonders steigern konnten, sah die Englischlehrerin mit als ausschlaggebend an. Zuletzt nannte die Mathematiklehrerin noch die Möglichkeit, etwas anderes tun zu dürfen, oder eben auch eine fehlende Alternativmöglichkeit.

Computer AG

Die Mathematik- und Deutschlehrerin stimmten darin überein, dass der Besuch der freiwilligen Computer AG wegen der zusätzlichen Übungszeit Vorteile für den Unterricht hatte (siehe Tabelle 6.9). Erstere behauptete, dass Inhalte in der Computer AG manchmal schon früher eingeführt worden seien. Dies geschah bei konkretem Nachfragen der Schülerinnen und Schüler. Inhaltlich handelte es sich um technische Fragen zur Bedienung des Logo-Systems und einige Logo-Befehle und nicht um informatische Grundkonzepte. Richtig ist sicherlich, dass das Engagement vieler Kinder, die an der Computer AG teilnahmen, auch im regulären Unterricht größer war. Dies lag nach Ansicht der Englischlehrerin jedoch nicht an der Computer AG an sich, sondern an der hohen Eigenmotivation der Kinder. Obwohl die Schülerinnen und Schüler während des

VIII, 5) Konnten Sie beobachten, dass die Schülerinnen und Schüler, die zusätzlich die Computer AG besuchten, einen Vorteil im normalen Computerunterricht hatten? In welchen Bereichen hatten sie Vorteile?	Anzahl
die Übung	2
vorher eingeführte Inhalte	1
persönlicher Einsatz der Schülerinnen und Schüler	1

Tabelle 6.9: Vorteile durch den Besuch der Computer AG (3 Lehrerinnen, Mehrfachnennungen möglich)

Unterrichtsversuchs vor allem bei der Bearbeitung der Aufgaben am Computer ihre Zeit frei und selbstbestimmt gestalten konnten, forderte die Englischlehrerin, den Anteil des Ausprobierens und Experimentierens zu erhöhen, damit vor allem auch die Kinder, die

die Computer AG nicht besuchten, mehr Möglichkeiten zum praktischen Üben hätten. Wichtig war ihr dabei das Verständnis und nicht das Entstehen von Produkten.

„Man müsste halt auch mehr Zeit haben und nicht einfach denken, ha ja, es muss doch was bei rauskommen, die Schüler müssen das verstehen.“ (LE, S. 10, 30 f.)

Zusammenfassung

Obwohl im Unterricht zusätzlich an den informatischen Grundkonzepten gearbeitet wird, können alle Fachinhalte der Klassenstufe 5 vermittelt werden. Im Englischunterricht unterstützt sogar die Beschäftigung mit den informatischen Grundkonzepten das Erlernen grammatikalischer Strukturen. Diese werden den Schülerinnen und Schülern mit Hilfe der verschiedenen informatischen Beschreibungsmittel deutlicher.

Aus Sicht der Lehrerinnen lernen die Kinder außerdem das Vorausdenken, das Strukturieren, das Anwenden von Schemata, das genaue Arbeiten, Strategien zur Bewältigung von Frustrationen und das Verstehen von Fehlermeldungen. Da es sich hier um Schlüsselqualifikationen handelt, die nicht allein für eine informationstechnische Grundbildung wichtig sind, sondern in allen Bereichen sowohl in der Schule als auch im späteren Beruf eine enorme Bedeutung haben, wird der Effekt des Unterrichtskonzepts bestätigt. Wichtig für diesen Erfolg ist die über ein Schuljahr stattfindende Arbeit an den informatischen Grundkonzepten, die systematisch eingeführt und in unterschiedlichen Fachkontexten wiederholt werden. Eine Benachteiligung durch das Unterrichtskonzept sehen die Lehrerinnen u.a. für Kinder mit schlechtem Abstraktionsvermögen. Diesen Kindern müsste insgesamt mehr Zeit zur Verfügung gestellt werden, damit sie das abstrakte Denken länger an einfachen Beispielen üben könnten.

6.1.4 Verhalten der Schülerinnen und Schüler im Unterricht

Weitere wichtige Indikatoren zur Beurteilung des Gesamtkonzepts sind die Motivation und Ausdauer, die Bedeutung, die der Unterricht und die Arbeiten für die einzelnen Schülerinnen und Schüler im Laufe des Schuljahrs hatten, und die Kooperation der Kinder untereinander.

Motivation

Die Lehrerinnen unterschieden zwischen positiven und negativen Motivationsfaktoren, die sich im Laufe des Schuljahrs änderten. Als positive Faktoren wurden die Möglichkeit, etwas für das Leben zu lernen, die Arbeit am Computer, das Gefühl, etwas Neues zu machen, Freude und Interesse genannt. Dem wurden von der Mathematik- und Englischlehrerin zum Beispiel die Haltung eines Elternpaares gegenübergestellt, die dem Forschungsprojekt lange Zeit sehr ablehnend gegenüberstanden. Dies wirkte sich auf das Kind dahingehend aus, dass es sich am Unterricht nicht beteiligte. Nach Meinung

der Englischlehrerin hatte auch die fehlende Vergabe von Noten einen negativen Einfluss auf die Motivation.

„[...] die Idee, dass es dafür keine Noten gibt und deshalb man sich da auch nicht anstrengen musste, dass das alles auch freiwillig ist.“ (LE, S. 10, 6 f.)

Außerdem wurden noch der Theorieunterricht, die fehlenden Vorstellungsmöglichkeiten, mangelnde Konzentrationsfähigkeiten und eine zu geringe Anerkennung der Leistungen durch die Forscherin genannt. Bis auf die fehlende Notenvergabe und den Theorieunterricht bezogen sich die negativen Einflussfaktoren lediglich auf einzelne Schülerinnen und Schüler und können nicht verallgemeinert werden.

Die Kinder selbst wurden nicht nach der Motivation, sondern nach ihrer Mitarbeit im Unterricht gefragt. Insgesamt beurteilten sie diese eher schlecht (siehe Tabelle 6.10).

II, 1) Hast du im Computerunterricht gut mitgearbeitet?	Anzahl
schlecht	2
mittelmäßig	3
gut	1

Tabelle 6.10: Mitarbeit im Unterricht (6 Schülerinnen und Schüler)

Das Fehlen von Noten spielte hierbei keine Rolle. Allein das Mädchen mit den besten Leistungen bezüglich der informatischen Grundkonzepte gab an, dass es sich bei der Vergabe von Noten noch mehr angestrengt hätte. Als negative Einflussfaktoren auf die Mitarbeit nannten die Kinder fehlende Motivation und Desinteresse am Unterricht. Die negative Beurteilung wurde von zwei Kindern auch auf Verständnisschwierigkeiten zurückgeführt. Im Zusammenhang mit den in Kapitel 6.1.2 beschriebenen Schwierigkeiten mit neuen Unterrichtsinhalten könnte die tendenziell schlechte Beurteilung der Mitarbeit größtenteils auf mangelndes Verständnis zurückgeführt werden.

Da die Motivation nicht unbedingt von den Fachinhalten oder der Lehrperson abhängig sein muss, sondern aus einer Haltung der Schülerinnen und Schüler gegenüber Schule und Unterricht im Allgemeinen resultieren kann, wurden die Lehrerinnen noch zu Unterschieden bezüglich der Motivation im Unterricht zu den informatischen Grundkonzepten und dem Fachunterricht gefragt. Zwei von ihnen sahen hier keine Unterschiede. Die Deutschlehrerin hob zweimal hervor, dass die Motivationsunterschiede nicht von der Lehrperson abhängig seien, sondern individuelle Gründe ausschlaggebend wären. Als Beispiel nannte sie einen Schüler, der bei der Computerarbeit sehr viel motivierter als in ihrem Unterricht war. Dies begründete sie damit, dass er im Deutschunterricht aufgrund von Rechtschreibproblemen große Schwierigkeiten hatte, seine kreativen Arbeiten am Computer jedoch von seinen Mitschülerinnen und Mitschülern sehr bewundert wurden. Andererseits gab es ihrer Meinung nach auch Kinder, die den Theorieunterricht zu den

informatischen Grundkonzepten nicht verstanden und deswegen im Deutschunterricht motivierter mitarbeiteten.

Als Motivationshöhepunkt gaben die Lehrerinnen das Ritterprojekt, das Musikprogramm, das Autospiele und die Unterrichtseinheiten zu Schuljahresbeginn an. Auf diesen Gebieten konnten die Schülerinnen und Schüler sehr viel praktisch arbeiten, ihre Kreativität einsetzen und direkt mit dem Logo-System über visuelle Ausgaben im Grafikbereich oder Ton kommunizieren. Das beschriebene Motivationshoch zu Beginn des Schuljahrs kann auf den Neuigkeitseffekt und die Aussicht, viel mit dem Computer arbeiten zu dürfen, zurückgeführt werden. Schnell wurde jedoch deutlich, dass diese Arbeit sehr anstrengend sein kann, hohe mentale Leistungen erfordert und theoretisches Wissen voraussetzt.

Ausdauer

Die Lehrerinnen stufen die Ausdauer im Unterricht zu den informatischen Grundkonzepten im Vergleich zum Fachunterricht als normal ein. Sie war ihrer Meinung nach sowohl abhängig von den einzelnen Kindern als auch von den Inhalten. Die Mathematiklehrerin bezeichnete sie vor allem in den langen Theoriephasen als sehr gut.

Bedeutung des Unterrichts

Das Verhalten der Schülerinnen und Schüler wurde auch dadurch beeinflusst, wie gerne sie am Unterricht zu den informatischen Grundkonzepten teilnahmen, was ihnen im Unterricht wichtig war, welchen Stellenwert ihre Arbeitsergebnisse für sie selbst hatten und welche Dinge ihnen nicht gefielen. Die beiden schwächeren Kinder gaben zum Beispiel an, dass sie den Unterricht zur Vermittlung der informatischen Grundkonzepte nicht gerne besucht hätten. Er habe ihnen keinen Spaß gemacht und sie gelangweilt. Alle anderen nahmen gerne am Computerunterricht teil, obwohl es auch ihnen manchmal langweilig oder auch zu schwierig war.

Als besonders wichtig bezeichneten alle Schülerinnen und Schüler das Rennautospiele, gefolgt vom Ritterprojekt, an dem sie über einen sehr langen Zeitraum frei und kreativ arbeiten durften. Die Arbeiten, die dabei entstanden, scheinen eine sehr viel geringere Bedeutung zu haben (vgl. Tabelle 6.11).

I, 2) Was hat dir im Computerunterricht besonders gut gefallen? Was war spannend?	Anzahl
Autospiele	2
Fadenbilder	1
Reime	1
Bilder	1
Imagine Logo	1
MSW Logo	1

Tabelle 6.11: Wichtige Arbeiten im Computerunterricht (6 Schülerinnen und Schüler, Mehrfachnennungen möglich)

Möglicherweise konnten die Kinder die Fragen, was ihnen im Computerunterricht wichtig war und welches ihre wichtigsten Arbeiten waren, auch nicht differenziert genug beantworten. Die Nennung ganz verschiedener Arbeiten bestätigt jedoch, dass eine Vielfalt an Arbeitsangeboten über ein Schuljahr hinweg bestehen muss. Dies ergibt sich automatisch bei der Einbindung verschiedener Fächer. Als weitere wichtige Tätigkeit wurden noch die Erstellung von Animationen und ganz allgemeine Dinge wie das Spielen, die Erstellung eigener Produkte, das Rechnen, das Zeichnen und das Programmieren aufgezählt. Für den informatisch leistungsstärksten Schüler waren alle Inhalte besonders wichtig. Er konnte hier keinen besonders hervorheben.

Keinen Gefallen hatten die beiden schwachen Kinder am Ritterprojekt. Dies lag jedoch vor allem an ihren Partnern. Das Mädchen musste mit einem Jungen zusammenarbeiten und der Schüler hatte einen dominanten Partner, bei dem er nur sehr wenig selbst machen durfte. Ein weiterer Schüler fand es schwierig, mit einem englischsprachigen Programm zu arbeiten und zwei bemängelten die Arbeit ohne Computer. Allerdings fiel zwei Kindern überhaupt nichts Negatives ein.

Kooperation

Die beiden schwachen Schülerinnen und Schüler gaben nicht nur im Ritterprojekt an, dass sie lieber alleine an einem Computer gearbeitet hätten. Dies lag auch daran, dass sie die Aufgaben am Rechner aufteilten und so selbst höchstens die Hälfte der Zeit im Unterricht aktiv waren. Die anderen hingegen sahen den Vorteil der Partnerarbeit in der Möglichkeit, sich gegenseitig zu helfen. Bei ihnen fand richtige Partnerarbeit und nicht nur abwechselndes Arbeiten statt. Beide waren gedanklich aktiv am Problemlösungsprozess beteiligt, auch wenn nur einer die Tastatur und die Maus bedienen konnte (vgl. Tabelle 6.12).

IV, 2) Habt ihr am Computer alles zusammen gemacht oder die Aufgaben eher aufgeteilt?	Anzahl
Zusammenarbeit	1
Aufteilung	2
Mischform	2
keine Aussage	1

Tabelle 6.12: Arbeitsformen am Computer (6 Schülerinnen und Schüler)

Die Englischlehrerin sah vor allem bei den Jungen der Klasse ein fehlende Kooperationsbereitschaft.

„Also bei den meisten Mädchenpaaren war das der Fall. Bei den Jungs war das etwas anders. Da hat wirklich dann derjenige, der nicht am Computer saß, auch geistig abgeschaltet.“ (LE, S. 11, 20 f.)

In diesem Fall verfehlt jedes Unterrichtskonzept, das die Zusammenarbeit von zwei Kindern an einem Computer vorsieht, einen Teil seiner Wirkung.

Nicht nur die Zusammenarbeit der Partner an einem Rechner, sondern auch die der Kinder im Klassenverband war insgesamt eher schlecht. Hilfestellungen im Unterricht hatten nur die Hälfte der interviewten Kinder gegeben, die Hilfe von anderen angenommen jedoch fast alle. Lediglich der beste Schüler sagte, dass er keine Hilfe von anderen gebraucht habe. Dies wurde auch dadurch deutlich, dass die Kinder bei Schwierigkeiten sehr stark auf die Forscherin fixiert waren. In etwa der Hälfte der Fälle wurde sie und nicht ein Mitschüler oder eine Mitschülerin gefragt (siehe Tabelle 6.13).

IV, 5) Hast du bei Problemen eher die Lehrerin oder eher deine Mitschülerinnen und Mitschüler zu Hilfe gerufen?	Anzahl
Lehrerin	2
Lehrerin oder Mitschülerinnen und Mitschüler	2
Mitschülerinnen und Mitschüler zuerst, dann die Lehrerin	1
Mitschülerinnen und Mitschüler, bei schwierigen Problemen die Lehrerin	1

Tabelle 6.13: Hilfe bei Schwierigkeiten (6 Schülerinnen und Schüler)

Das Ziel, die Kommunikation der Kinder untereinander über ihre Produkte zu fördern, wurde also nicht erreicht.

Zusammenfassung

Die Motivation der Schülerinnen und Schüler im Unterricht zu den informatischen Grundkonzepten ist normal. Wie auch in anderen Fächern hängt sie vor allem von den individuellen Interessen und Fähigkeiten der Kinder und weiteren Faktoren wie zum Beispiel den Einstellungen der Eltern, der Vergabe von Schulnoten, der Anerkennung der Leistungen durch die Lehrperson, dem Einsatz des Mediums Computer und dem Neuigkeitseffekt zu Beginn des Schuljahrs ab. Motivationshöhen gibt es in den Unterrichtseinheiten, in denen sehr viel praktisch kreativ gearbeitet wird. Die Ausdauer dagegen wird teilweise in zu langen Theoriephasen überstrapaziert und die Mitarbeit von den Schülerinnen und Schülern selbst als schlecht eingestuft, wenn die Inhalte des Unterrichts nicht sofort verstanden werden. Die Kinder fühlen sich dann überfordert und sind gelangweilt. Sehr problematisch ist die teilweise schlechte Zusammenarbeit der Partner an den Computern. Vor allem bei den Jungen ist nur derjenige Partner geistig aktiv bei der Arbeit, der auch die Tastatur und Maus bedient. Eine Kommunikation zwischen Schülerinnen und Schülern, die nicht zusammenarbeiteten, findet kaum statt.

6.1.5 Rahmenbedingungen

Zwei Lehrerinnen in einem Fach

Eine echte Integration der informatischen Grundkonzepte in den Fachunterricht kann nach Meinung der Mathematik- und Deutschlehrerin erst erreicht werden, wenn der Unterricht zu den informatischen Grundkonzepten auch von den Fachlehrerinnen selbst durchgeführt wird. Als ein Nachteil wurde zum Beispiel die teilweise fehlende Übereinstimmung der Begrifflichkeiten genannt.

Zwei Arbeitsräume

Einen sehr großen Einfluss auf die Arbeit hatte die Tatsache, dass in zwei verschiedenen Räumen gearbeitet wurde (vgl. Tabelle 6.14). Von zwei Lehrerinnen wird die Bewegung

III, 3) Solange wir die Computer im Computerraum verwendet haben, fand ein Teil des Computerunterrichts im Klassenzimmer statt. Für die Computerarbeit musste die Klasse dann in den Computerraum wechseln. Wie beurteilen Sie die Aufteilung zwischen der Arbeit im Klassenzimmer und im Computerraum? – Vorteile		Anzahl
Bewegung		2
beim Erklären sitzen die Kinder nicht vor Computern		1
Schülerinnen und Schüler sitzen an ihren gewöhnlichen Plätzen		1
III, 3) Nachteile		Anzahl
Theorie im Klassenzimmer – Praxis im Computerraum		2
Plakate nur im Computerraum		1
Tafelanschrieb im Klassenzimmer fehlt im Computerraum		1

Tabelle 6.14: Vor- und Nachteile des Raumwechsels (3 Lehrerinnen, Mehrfachnennungen möglich)

positiv bewertet, die die Schülerinnen und Schüler zwangsläufig beim Wechsel des Raums haben. Bewegung kann jedoch auch in einem Unterrichtsraum durch geeignete methodische Maßnahmen gefördert werden. Außerdem kann das Sitzen am eigenen Platz im Klassenzimmer auch als nachteilig empfunden werden, da einige Arbeiten im Computerraum vorbereitet wurden und die Tischnachbarn im Klassenzimmer nicht unbedingt Partner im Computerraum waren. Wie bereits oben angesprochen muss die Verknüpfung des Klassenzimmers mit theoretischen und des Computerraums mit praktischen Arbeiten auf jeden Fall vermieden werden. Hier dürfen die Kinder auf keinen Fall eine Polarisierung in unangenehme und angenehme Unterrichtsstunden erleben. Vielmehr muss ihnen bewusst werden, dass erlernte theoretische Konzepte hilfreich für eine effiziente Arbeit am Computer sind. Einen weiteren großen Nachteil stellten die Unterrichtsmaterialien dar, die nur in einem der Unterrichtsräume vorhanden waren. Die Lehrerinnen nannten hier die Plakate, auf denen die eingeführten Logo-Befehle standen, oder die im Computerraum fehlenden Tafelanschriebe aus dem Klassenzimmer. Dem hinzuzufügen ist noch

das Vergessen der Hefte im Klassenzimmer, so dass einzelnen Schülerinnen und Schülern im Computerraum dann weitere Arbeitsmaterialien fehlten.

Doppelstunden

Prinzipiell waren nach Meinung der Lehrerinnen Doppelstunden für die Vermittlung der informatischen Grundkonzepte geeignet. Sie hoben jedoch hervor, dass an theoretischen Themen nicht über zwei Stunden hinweg gearbeitet werden darf. Für die Arbeit am Computer wurden Doppelstunden dagegen als wichtig eingeschätzt.

Partnerarbeit am Computer

Da im Computerraum nur 16 Rechner zur Verfügung standen, mussten jeweils zwei Kinder zusammenarbeiten. Dies ermöglichte ihnen, sich gegenseitig zu helfen, was von zwei Lehrerinnen als Vorteil angesehen wurde. Richtigerweise argumentierte die Englischlehrerin, dass dies auch mit Kindern an benachbarten Computern möglich sei. Sie kritisierte an der Partnerarbeit, dass nicht beide Partner aktiv seien. Dies beobachtete sie vor allem bei den Jungen. Wie bereits in Kapitel 6.1.4 dargestellt, entspricht dies der Aussage der schwächeren Kinder, die die Zusammenarbeit mit einem Partner an einem Rechner selbst als problematisch einschätzten.

Hausaufgaben

Hausaufgaben zur Vermittlung der informatischen Grundkonzepte wurden nur teilweise als notwendig erachtet. Die Mathematik- und Deutschlehrerin waren überzeugt, dass die theoretischen Inhalte auch zu Hause geübt werden müssten. Sie forderten gleichzeitig eine Bewertung, damit sie für die Schülerinnen und Schüler einen höheren Stellenwert bekämen. Die Englischlehrerin sah das Üben im Unterricht als ausreichend an.

Logo

Wie die Lehrerinnen die Auswirkungen des ausschließlichen Einsatzes von Logo-Systemen einschätzten, wurde schon in Kapitel 6.1.2 diskutiert. Die Englisch- und Deutschlehrerin sahen durch die Eingrenzung Vorteile. Die Schülerinnen und Schüler selbst fühlten sich nicht eingeschränkt. Auf die Frage, ob sie etwas beim Arbeiten mit dem Logo-System gestört habe, beklagten sich lediglich zwei von ihnen, dass die Befehlsnamen in Imagine Logo englisch gewesen seien und sie die entsprechenden Wörter nicht im Englischunterricht gelernt hätten (vgl. Tabelle 6.15).

Vielmehr beschrieben sie eine höhere Flexibilität bei der Arbeit mit Logo im Vergleich zu Standardprogrammen und Spielen. Ihrer Meinung nach waren die Möglichkeiten der Gestaltung in Logo größer. Außerdem konnten sie durch Programmierung ihre eigenen Ideen umsetzen. Was mit der Speicherung gemeint wurde, kann nicht nachvollzogen werden.

Interessant ist außerdem, welche Unterschiede den Schülerinnen und Schülern zwischen MSW Logo und Imagine Logo auffielen (vgl. Tabelle 6.16). Offensichtlich waren dies die Sprache, in der die Befehle eingegeben werden mussten, der Igel, der eigentlich nur als Dreieck dargestellt wurde, und die Schildkröte und schließlich die Möglichkeit der Ver-

VI, 1) Gab es insgesamt etwas, was dich bei der Arbeit mit Logo gestört hat?	Anzahl
nichts	2
englische Befehle	2
VI, 3) Vergleiche nun deine Arbeit mit Logo mit der Arbeit mit den anderen Programmen. Was war an der Arbeit mit Logo besonders für dich?	Anzahl
mehr Gestaltungsmöglichkeiten im Vergleich zu Textverarbeitungsprogrammen	2
Gestaltung eigener Spiele	1
Programmierung eigener Dinge	2
Speicherung	1

Tabelle 6.15: Vergleich von Logo und anderen Programmen (6 Schülerinnen und Schüler)

VI, 4) Kannst du den Unterschied zwischen MSW Logo und Imagine Logo beschreiben?	Anzahl
Deutsch – Englisch	4
Igel – Schildkröte	2
ein Igel – viele Tiere	1
keine Gestaltungsmöglichkeit – Gestaltung	4
– Animation von Objekten durch Programmierung	1
– wie gewohnte Programme	1
kaum ein Unterschied erkennbar	1

Tabelle 6.16: Vergleich von MSW Logo und Imagine Logo (6 Schülerinnen und Schüler, Mehrfachnennungen möglich)

wendung eines Igels oder vieler Tiere. Erstaunlich war die Unterscheidung der Systeme durch die Mehrheit der Kinder dahingehend, dass nur mit Imagine Logo gestaltet werden konnte, da auch in MSW Logo zum Beispiel Fadenbilder, Weihnachtskarten oder andere Bilder entwickelt wurden.

Der Begriff der Gestaltung wurde von den Schülerinnen und Schülern wohl im Zusammenhang der Erstellung multimedialer Produkte verwendet. Außerdem konnten nach Ansicht eines Schülers in Imagine Logo Objekte durch Programmierung animiert werden. Eine Schülerin beschrieb die Ähnlichkeit von Imagine Logo und Standardanwendungen, die sie bereits kannte². Es wird deutlich, dass die Aufgabenstellungen und die Möglichkeit der einfachen Einbindung von Texten und Bildern von dem Kern der Arbeit, der Formalisierung von Sachverhalten mit Hilfe von eigenen Programmen, ablenkten.

²Dies bezog sich sicherlich auf die Auswahl von Funktionen aus der Toolbar, den Aufbau des Kontextmenüs zur Veränderung der Objekteigenschaften, das Zeichenprogramm, die Möglichkeit, Texte und Bilder einzubauen, und den farbigen Aufbau des Programms.

Weitere Rahmenbedingungen

Die Lehrerinnen beschrieben noch die Lage der Schulstunden am frühen Morgen, die Ausstattung des Computerraumes, die Projektpräsentationen vor den Eltern und die am Ende des Schuljahrs versprochene Belohnung für diejenigen, die ihre Leistung bezüglich der informatischen Grundkonzepte am meisten steigerten, als positive Einflussfaktoren.

Zusammenfassung

Insgesamt gesehen haben die verschiedenen Rahmenbedingungen keinen entscheidenden Einfluss auf den Erfolg des Unterrichtskonzepts. Eine richtige Integration der informatischen Grundkonzepte in den Fachunterricht kann nach Meinung der Lehrerinnen erst erfolgen, wenn sie diese selbst unterrichten. Die Arbeit in zwei verschiedenen Räumen wirkt sich dann nachteilig aus, wenn die Schülerinnen und Schüler eine Polarisierung in angenehme praktische und unangenehme theoretische Stunden wahrnehmen oder die notwendigen Unterrichtsmaterialien wie Plakate und Tafelanschriften nicht zur Verfügung stehen. Theoretische Fragestellungen dürfen nicht über Doppelstunden hinweg bearbeitet werden. Hausaufgaben sind auch in diesem Unterrichtsarrangement notwendig, um die theoretischen Inhalte zu üben. Noch mehr muss auf die Möglichkeiten in Logo aufmerksam gemacht werden, dass im Gegensatz zu Spielen oder Standardanwendungen Programme aktiv gestaltet und den eigenen Wünschen angepasst werden können.

6.2 Projektarbeit

Mit der Arbeit am Ritterprojekt veränderte sich das methodische Konzept des Unterrichts. Die informatischen Grundkonzepte wurden bei der fächerübergreifenden Arbeit angewendet. Nach Altermann-Köster & al. (1990, S. 73) kann davon ausgegangen werden, dass ein projektorientierter Grundbildungsunterricht dann qualitativ hochwertig ist und von den Schülerinnen und Schülern akzeptiert wird, wenn die Schülerinnen und Schüler zu folgenden Aspekten positive Beurteilungen abgeben:

- Lebensweltbezug der Lerninhalte
- fächerübergreifender Projektcharakter
- handlungsorientierte Eigentätigkeit
- gesellschaftsorientiertes Lernen
- Einsatz neuer Technologien.

Der Erfolg des Ritterprojekts wird in der Zusammenfassung des Kapitels auf Basis der Aussagen der Schülerinnen und Schüler und der Rahmenbedingungen des Projekts begründet.

Rahmenbedingungen

Die Projektarbeit begann nach etwa sieben Monaten Unterricht zu den informatischen Grundkonzepten. Die für die Rittergeschichte zur Verfügung gestellte Zeit bewertete die

Mehrheit der Lehrerinnen als gut. Die Deutschlehrerin hätte die Anzahl der Stunden etwas begrenzt und das Projekt stärker strukturiert. Darin stimmten auch fast alle Schülerinnen und Schüler überein. Die schwächeren Kinder hätten nichts zusätzlich umsetzen wollen. Die anderen nannten Animationen und Bilder, die ohnehin schon Bestandteil der Projektarbeit waren. Allein der Wunsch, Audiodaten aufzunehmen und als Dateien in das Projekt einzubinden, war ein neuer Inhalt.

Das Thema wurde sowohl für Mädchen als auch für Jungen als geeignet oder sogar ideal eingeschätzt. Die Rahmenbedingungen bildeten also die Grundlage für eine gute Projektarbeit.

Projekterfolg

Die Lehrerinnen beschrieben die Einstellungen und Gefühle der Kinder während des Projektunterrichts sehr positiv (vgl. Tabelle 6.17).

IV, 1) Sehen Sie das Ritterprojekt als erfolgreich an? Wie beurteilen Sie das Projekt?	Anzahl
IV, 5) Wie sind die Schülerinnen und Schüler mit den Freiheiten der Projektarbeit zurechtgekommen?	
Einstellungen	
motiviert	1
konzentriert	2
normal bezüglich der Mitarbeit	1
identifizierend	1
Gefühle	Anzahl
keine Überforderung	1
großer Spaß	1
Euphorie	1
Bewusstsein, etwas Besonderes zu machen	1

Tabelle 6.17: Einstellungen und Gefühle während der Projektarbeit (3 Lehrerinnen, Mehrfachnennungen möglich)

Zusätzlich hoben die Lehrerinnen hervor, dass alle Kinder die Möglichkeit hatten, sich mit ihren ganz speziellen Fähigkeiten einzubringen. Dadurch seien die tollen fächerübergreifenden Leistungen erst möglich gewesen. Ohne nähere Spezifikation sagten sie, dass die Schülerinnen und Schüler während des Projekts viel gelernt hätten. Die Projektergebnisse wurden als imponierend, erstaunlich und „gigantisch“ bezeichnet.

Die Kinder stellten die Arbeit an den Laptops, die sehr freie Arbeitsform, den Unterricht ohne Arbeitsblätter und einzelne Inhalte des Ritterprojekts als das Besondere an der Arbeit heraus. Demgegenüber beschrieben drei Schülerinnen und Schüler den Partner, mit dem sie zusammenarbeiten mussten, das Eintippen der Rittergeschichte und die

Anstrengung, sich einiges Ausdenken zu müssen, als etwas Negatives. Letztere Aussage machte jedoch ein Schüler, dem gerade das Ausdenken seiner Geschichte und das Malen der Bilder sehr viel Spaß gemacht hatte. Das Gesagte muss dementsprechend relativiert werden.

Lerninhalte

Das Lernen der informatischen Grundkonzepte fand nach Meinung der Mathematik- und Englischlehrerin vor dem Ritterprojekt statt. Erstere ging sogar soweit, dass sie keine Möglichkeit sah, diese in Projekten zu vermitteln. Die andere schränkt dies unter der Bedingung ein, dass sehr viel Zeit und extrem viel Geduld notwendig wären. Die Deutschlehrerin sah sogar die Projektarbeit als geeigneter als den Theorieunterricht an, schlug jedoch auch eine Mischform vor.

Inhaltlich sahen die Schülerinnen und Schüler bis auf die Tatsache, dass an verschiedenen Inhalten in einem Produkt gearbeitet wurde, keine Unterschiede zwischen der Projektarbeit und dem Unterricht zuvor. Sie nahmen den Wechsel vom Erlernen hin zur Anwendung der informatischen Grundkonzepte überhaupt nicht wahr. Viel auffälliger waren für sie methodische Veränderungen wie das selbstständige Arbeiten, die Realisierung eigener Ideen, die reduzierten Hilfen und die Arbeit an einem umfangreichen Produkt über einen längeren Zeitraum hinweg. Von einem Schüler wurde die Projektarbeit auch als schwieriger eingestuft. Ein Kind erwähnte außerdem die Arbeit an den Laptops. Diese konnten jedoch eher zufällig zu Beginn der Projektarbeit zur Verfügung gestellt werden.

Schwierigkeiten

Wichtig ist festzuhalten, dass nicht alle Aufgaben im Ritterprojekt in der vorgesehenen Art und Weise gelöst wurden. Beim Zeichnen der achsengespiegelten Burg gaben sich alle Schülerinnen und Schüler mit der ursprünglichen Form zufrieden (vgl. Kapitel 4.2, S. 93). Gleiches galt für die generierten Gedichte. In keinem der beiden Fälle wurde experimentiert. Als Gründe dafür gaben die Kinder ihre Faulheit, Schwierigkeiten bei der Entwicklung anspruchsvollerer Lösungen und ihr knappes Zeitbudget an, das sie lieber in anderen Bereichen investierten. Zwei von ihnen waren mit der einfachen Lösung zufrieden und einer Schülerin reichte es aus, genau das zu tun, was die Lehrerin von ihr verlangte.

Die Vermutungen der Lehrerinnen stimmten mit den Begründungen der Schülerinnen und Schüler sehr gut überein (siehe Tabelle 6.18). Der Faulheit kann als Begründung die nicht erbrachte zusätzliche Anstrengung zugeordnet werden. Außerdem nannten die Lehrerinnen auch alternative Arbeitsmöglichkeiten. Die Gleichgültigkeit entspricht dem Vorgehen, genau das zu machen, was die Lehrperson vorgibt. Ein neuer Aspekt hingegen ist die von der Englisch- und Deutschlehrerin vermutete fehlende Vorstellungskraft für Burgen aus geometrisch interessanteren Grundformen.

Lehrerinnen (IV, 6)	Anzahl	Schülerinnen und Schüler (VII, 6)	Anzahl
zusätzliche Anstrengung	3	Faulheit	1
alternative Freiheiten	1	enges Zeitbudget	5
Gleichgültigkeit	1	Abarbeiten der Vorgaben der Lehrerin; Zufriedenheit mit der einfachen Lösung	3
fehlende Vorstellung	2		
		schwierige Lösung	2

Tabelle 6.18: Ursache für fehlende Experimentierfreudigkeit (3 Lehrerinnen, 6 Schülerinnen und Schüler, Mehrfachnennungen möglich)

Wie schon in anderen Unterrichtseinheiten war auch im Projektunterricht in einigen wenigen Fällen die Zusammenarbeit am Computer problematisch. Vor allem bei den im Vergleich zu den sonstigen Arbeiten sehr kreativen Tätigkeiten hätten einige Kinder gerne selbst mehr gemacht. Allerdings gab es auch einige Partner, die sich optimal ergänzten. Der eine übernahm dann bei den eher technisch orientierten Programmiertätigkeiten und der andere in den kreativ geprägten Teilen der multimedialen Rittergeschichte die Führung. Optimal wäre es also, wenn genügend Computer zur Verfügung stehen würden, so dass sich jede Schülerin und jeder Schüler zwischen Einzel- und Partnerarbeit entscheiden und im letzteren Fall auch seinen Partner auswählen könnte.

Zusammenfassung

Nach Altermann-Köster & al. (1990, S. 73) war das Projekt der multimedialen Rittergeschichte sehr erfolgreich. Während der Projektarbeit wurden neue Technologien eingesetzt. Wie die Schülerinnen und Schüler selbst feststellten, arbeiteten sie an mehreren Inhalten gleichzeitig, also fächerübergreifend. Auch die Handlungsorientierung war so deutlich, dass die Kinder sie in den Interviews nannten. Da es bei der Einbeziehung der Fachinhalte in das Ritterprojekt darum ging, diese in einem Produkt praktisch umzusetzen und anzuwenden, kann auch von einem Bezug zur Lebenswelt gesprochen werden. Die Lehrerinnen sahen das Thema der Rittergeschichte sowohl für Mädchen als auch für Jungen als geeignet an. Allein für interessantere Grundrisse von Burgen fehlte den Kindern die Vorstellungskraft. Die Lehrerinnen betonten außerdem, dass sich alle Kinder mit ihren ganz speziellen Fähigkeiten einbringen konnten und dabei eine tolle fächerübergreifende Leistung erzielten. Die entstandenen multimedialen Rittergeschichten beschrieben sie als imponierend, erstaunlich und „gigantisch“. Allein das gesellschaftsorientierte Lernen fand im Projekt nicht statt. Allerdings wurde es im Laufe des gesamten Schuljahrs immer wieder thematisiert.

6.3 Lernerfolg informatischer Grundkonzepte – Vor- und Nachtest

Die Lehrerinnen, Schülerinnen und Schüler gaben, wie in Kapitel 6.1.3 dargestellt, keine exakte Auskunft über den Lernerfolg bezüglich der informatischen Grundkonzepte. Ein Grund dafür war, dass sie die Inhalte im Unterricht nicht als informatische Grundkonzepte wahrgenommen hatten. Aus diesem Grund erfolgte eine Quantifizierung der Konzepte mit Hilfe des in Kapitel 5.2.2 beschriebenen Tests.

Der *Lernerfolg* wurde als Differenz der Testergebnisse am Ende und zu Beginn des Schuljahrs definiert. In Kapitel 6.3.1 wird die Punkteverteilung der Schülerinnen und Schüler im Vor- und Nachtest dargestellt. Außerdem wurde mit dem Wilcoxon-Test zum Vergleich zweier abhängiger Stichproben bezüglich ihrer zentralen Tendenzen der Lernerfolg auf Signifikanz geprüft.

Anschließend wird in den Kapiteln 6.3.2 und 6.3.3 der Lernerfolg für die Merkmale

- Geschlecht,
- kognitives Gesamtleistungsniveau,
- sprachgebundenes Denken,
- zahlengebundenes Denken und
- formallogisches Denken

mit den Merkmalsausprägungen „Junge“, „Mädchen“, „stark“ und „schwach“ untersucht. Die Darstellung der Ergebnisse erfolgt in Box-Plots, in denen die minimalen und maximalen Lernerfolge, die Mediane als zentrale Tendenzen der Verteilungen der Ergebnisse und die Streuungen in Form der Interquartilabstände Q1 und Q3 gekennzeichnet sind. Außerdem wurde der Unterschied des Lernerfolgs bezüglich der korrespondierenden Ausprägungen jedes Merkmals mit Hilfe des U-Test von Mann und Whitney zum Vergleich von zwei Stichproben hinsichtlich ihrer zentralen Tendenz auf Signifikanz überprüft.

Die Rohdaten der Auswertung des Leistungstests zu den informatischen Grundkonzepten und des KFT (*Kognitiver Fähigkeitstest*), mit dem die Fähigkeiten *kognitives Gesamtleistungsniveau*, *sprachgebundenes Denken*, *zahlengebundenes Denken* und *formallogisches Denken* gemessen wurden, sind in Tabelle 6.19 dargestellt. Die Nullwerte sind aufgrund der in Heller & al. (2000) beschriebenen Auswertungsanweisungen entstanden, da die Schülerinnen und Schüler in verschiedenen Teilen des Tests nur sehr wenige Punkte erzielt hatten.

Kind	m/w	Informatische Grundkonzepte		Kognitiver Fähigkeitstest (KFT)			
		Beginn	Ende	gesamt	sprachgeb.	zahlengeb.	formallog.
1	m	13	14	63	48	69	63
2	m	11	17	50	39	44	65
3	m	7	17	49	30	53	73
4	m	7	14	50	35	61	61
5	m	12	19	48	29	52	60
6	m	8	15	43	0	49	60
7	m	5	11	35	27	43	47
8	m	8	14	38	27	53	45
9	m	7	10	49	36	40	69
10	m	3	12	49	40	41	55
11	m	7	17	38	25	33	53
12	m	6	12	34	27	33	50
13	m	3	12	0	28	0	32
14	w	6	12	53	39	52	71
15	w	9	10	52	38	55	60
16	w	9	10	48	52	49	45
17	w	6	8	45	35	52	49
18	w	10	18	57	54	41	73
19	w	8	18	49	47	39	58
20	w	4	8	49	45	40	55
21	w	9	6	46	45	40	54
22	w	10	11	57	60	38	60
23	w	15	16	47	51	33	53
24	w	4	8	43	47	27	53
25	w	9	10	42	40	35	54
26	w	9	10	45	52	28	52
27	w	8	8	46	47	37	50
28	w	10	10	43	55	29	46
29	w	9	10	38	40	0	53
30	w	4	5	30	32	30	36

Tabelle 6.19: Rohdaten der Tests zu den informatischen Grundkonzepten und des KFT

6.3.1 Informatische Grundkonzepte allgemein

Die Häufigkeitsverteilungen der erreichten Punkte im Vor- und Nachtest zu den informatischen Grundkonzepten sind in Abbildung 6.3 dargestellt.

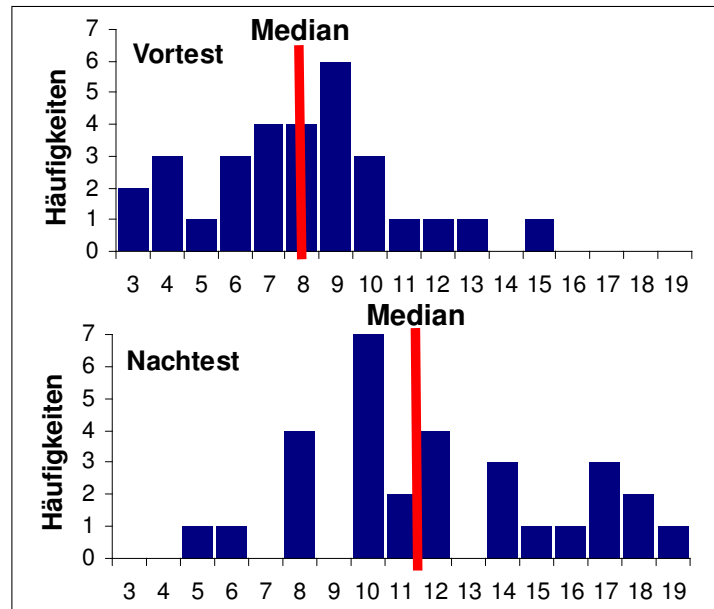


Abbildung 6.3: Häufigkeitsverteilung der Punkte im Vor- und Nachtest

Im Nachtest ist eine deutliche Verschiebung der Häufigkeitsverteilung hin zu höheren Punktzahlen und somit besseren Leistungen zu erkennen. Abbildung 6.4 zeigt die genaue Leistungssteigerung vom Vor- zum Nachtest. Ein Kind verschlechterte sein Ergebnis, und zwei hatten in beiden Tests dieselbe Punktzahl.

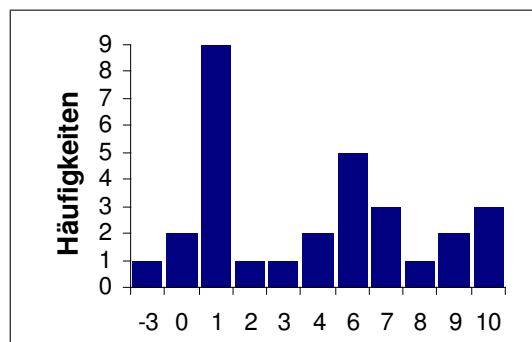


Abbildung 6.4: Häufigkeitsverteilung der Differenzen der Punkte des Nach- und Vortests

Signifikante Unterschiede zwischen dem Vor- und Nachtest wurden mit dem Wilcoxon-Test zum Vergleich zweier abhängiger Stichproben bezüglich ihrer zentralen Tendenzen überprüft. Das Alpha-Niveau von $p = 0,05$ wurde mit Hilfe der Bonferroni-Korrektur auf

$p = 0,05/6$ verschärft, da wie in den Kapiteln 6.3.2 und 6.3.3 beschrieben wird, insgesamt sechs Tests durchgeführt wurden (vgl. Zöfel 2002, S. 69). Daraus ergab sich für die Untersuchung ein Alpha-Niveau von $p = 0,008$. Da die berechnete Wahrscheinlichkeit kleiner als das festgelegte Alpha-Niveau von 0,008 ist ($Z = -4,382, p < 0,001$), kann die Nullhypothese, dass sich die Ergebnisse im Vor- und Nachtest nicht signifikant voneinander unterscheiden, verworfen werden.

Zusammenfassung und Interpretation

Bereits vor dem Unterricht zu den informatischen Grundkonzepten konnten die Schülerinnen und Schüler einige informatische Strategien anwenden. Der Unterschied im Vor- und Nachtest ist zwar signifikant, allerdings liegt er bei fast der Hälfte der Kinder unter vier Punkten. Wie auch in anderen Fächern hatten die Schülerinnen und Schüler am Ende des Schuljahrs einige informatische Inhalte wieder vergessen. Außerdem war ihre Konzentration im Nachtest am Schuljahresende vermutlich geringer. Sie waren mit der Situation bereits vertraut und ein Teil von ihnen glaubte, sich weniger anstrengen zu müssen.

6.3.2 Informatische Grundkonzepte und Geschlecht

Wie die beiden Box-Plots in Abbildung 6.5 zeigen, unterschied sich der Lernerfolg, d.h. die Differenz der Ergebnisse des Nach- und Vortests, bei den Jungen und Mädchen deutlich.

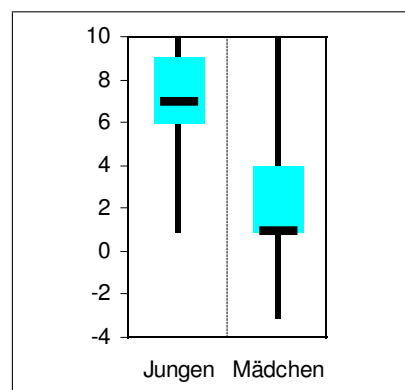


Abbildung 6.5: Box-Plots der Differenzen der Punkte des Nach- und Vortests – Geschlecht

Zwar lag die maximale Leistungssteigerung in beiden Gruppen bei zehn Punkten, im Gegensatz zu den Jungen verschlechterte sich aber ein Mädchen. Der Median der Jungen befindet sich bei den Schülern bei sieben und bei den Schülerinnen nur bei einem Punkt. Die Interquartilbereiche sind bei beiden Gruppen gleich groß, überlappen sich jedoch nicht.

Zu Beginn des Schuljahrs waren die Mädchen im Test zu den informatischen Grundkonzepten etwas erfolgreicher (vgl. Abbildung 6.6). Im Nachtest änderte sich dies. Unterhalb des Medians lagen 76 Prozent der Mädchen und nur 15 Prozent der Jungen.

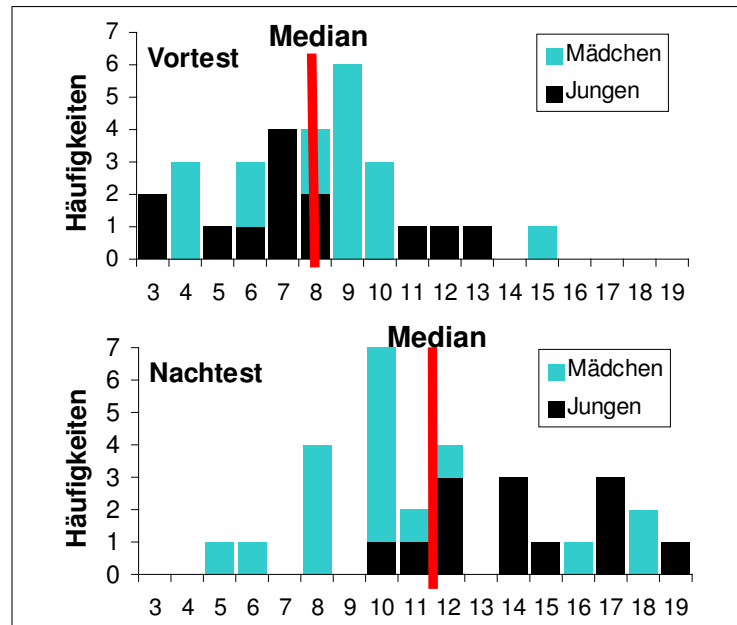


Abbildung 6.6: Häufigkeitsverteilung der Punkte im Vor- und Nachtest getrennt nach dem Merkmal *Geschlecht*

Die Ausgangssituation war also für beide Gruppen etwa gleich, die Entwicklung der Jungen jedoch wesentlich erfolgreicher. Die Nullhypothese, dass sich die Differenz der Ergebnisse des Nach- und Vortests der Jungen und Mädchen nicht signifikant voneinander unterscheidet, muss verworfen werden. Mit dem U-Test von Mann und Whitney wurde eine Wahrscheinlichkeit ($Z = -3,257, p = 0,001$) kleiner als das festgelegte Alpha-Niveau von 0,008 berechnet.

Zusammenfassung

Sowohl Mädchen als auch Jungen können im Alter von zehn Jahren informatische Grundkonzepte in schriftlichen Aufgaben ohne vorherigen speziellen Unterricht selbstständig anwenden. Nach einem Jahr Unterricht zu den informatischen Grundkonzepten erzielen die Jungen sehr viel bessere Testergebnisse. Die Leistungssteigerungen vom Vor- zum Nachtest sind bei ihnen sogar signifikant höher als bei Mädchen.

6.3.3 Informatische Grundkonzepte und Fähigkeiten

Mit Hilfe des KFT (*Kognitiver Fähigkeitstest*) wurden die verschiedenen Fähigkeiten der Schülerinnen und Schüler getrennt nach den Merkmalen

- kognitives Gesamtleistungsniveau,
- sprachgebundenes Denken,
- zahlengebundenes Denken und
- formallogisches Denken

erfasst. Für jedes Merkmal wurden die mittleren 20 Prozent der Schülerinnen und Schüler herausgefiltert, so dass sich in der Gruppe der Leistungsstarken und Leistungsschwachen jeweils 12 Kinder befinden. Die Verteilung der Jungen und Mädchen auf die verschiedenen Gruppen ist in Tabelle 6.20 dargestellt.

	<i>Absolute (relative) Häufigkeiten</i>			
	Jungen		Mädchen	
	<i>kognitives Gesamtleistungsniveau</i>			
leistungsschwach	6	(46,2%)	6	(35,3%)
mittel	1	(7,7%)	5	(29,4%)
leistungsstark	6	(46,2%)	6	(35,3%)
	<i>sprachgebundenes Denken</i>			
leistungsschwach	10	(76,9%)	2	(11,8%)
mittel	2	(15,4%)	4	(23,5%)
leistungsstark	1	(7,7%)	11	(64,7%)
	<i>zahlengebundenes Denken</i>			
leistungsschwach	3	(23,1%)	9	(52,9%)
mittel	2	(15,4%)	4	(23,5%)
leistungsstark	8	(61,5%)	4	(23,5%)
	<i>formallogisches Denken</i>			
leistungsschwach	5	(38,5%)	7	(41,2%)
mittel	1	(7,7%)	5	(35,3%)
leistungsstark	7	(53,8%)	5	(29,4%)

Tabelle 6.20: Verteilung der Mädchen und Jungen in den untersuchten Gruppen

In den Auswertungen wurden nur die leistungsschwachen und leistungsstarken Schülerinnen und Schüler bezüglich der vier Merkmale betrachtet. Zum Vergleich der korrespondierenden Gruppen wurden Box-Plots erstellt und Signifikanzen überprüft.

Die Box-Plots der Leistungssteigerungen der im KFT starken und schwachen Schülerinnen und Schüler in Abbildung 6.7 ganz links sind sich sehr ähnlich. Der Interquartilbereich, der Median und der Minimalwert liegen bei der Gruppe mit dem stärkeren kognitiven Gesamtleistungsniveau etwas höher.

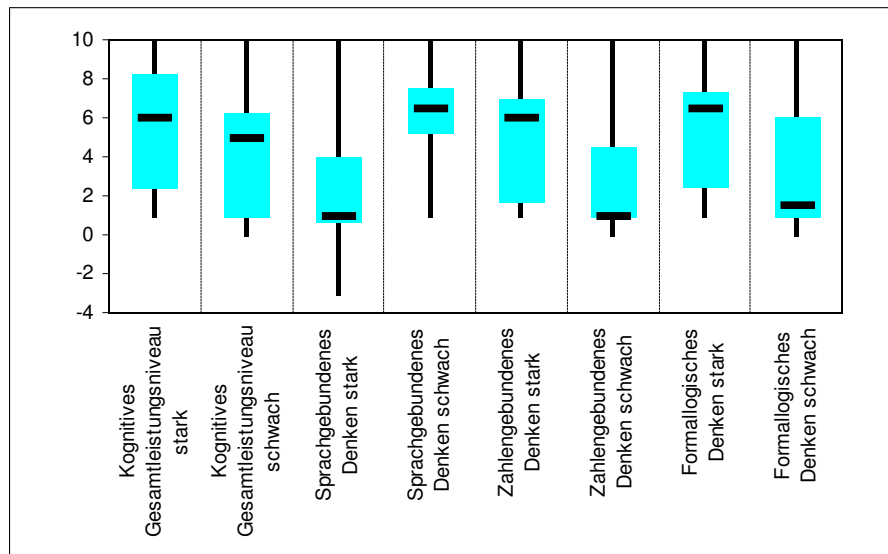


Abbildung 6.7: Box-Plots der Differenzen der Punkte des Nach- und Vortests – Fähigkeiten

Zu Beginn des Schuljahrs waren die Ergebnisse der beiden Gruppen im Test zu den informatischen Grundkonzepten sehr ausgeglichen um den Median verteilt (siehe Abbildung 6.8).

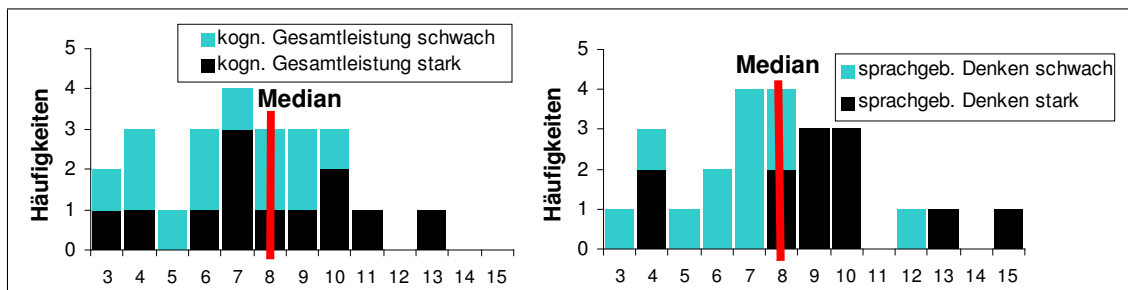
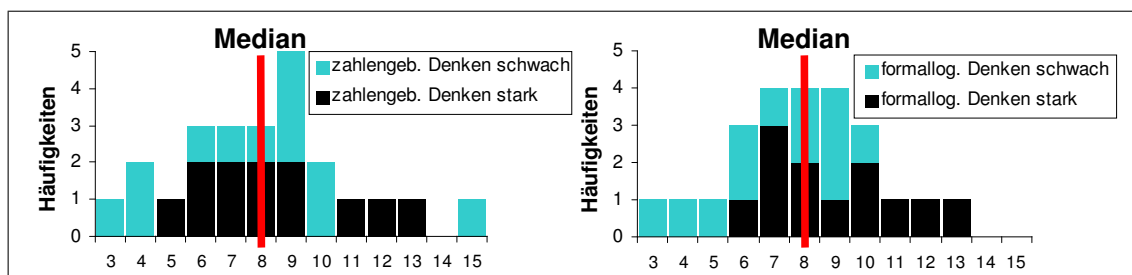


Abbildung 6.8: Häufigkeitsverteilungen der Punkte im Vortest getrennt nach den Merkmalen *kognitives Gesamtleistungsniveau* und *sprachgebundenes Denken*

Beim Merkmal *sprachgebundenes Denken* ist derselbe Effekt wie beim *Geschlecht* zu beobachten. Die im sprachgebundenen Denken schwächeren Schülerinnen und Schüler steigerten ihre Leistungen im Nachtest im Vergleich zum Vortest wesentlich mehr als die Kinder mit stärkeren Fähigkeiten in diesem Bereich. Die Mediane liegen sehr weit auseinander und es existiert keine Überlappung der Interquartilbereiche. Ein Grund für die Ähnlichkeit der Ergebnisse ist die Zusammensetzung der beiden Gruppen. In der Gruppe der im sprachgebundenen Denken schwachen Kinder ist der Anteil der Jungen wesentlich höher. Umgekehrt sind in der Gruppe der Stärkeren hauptsächlich Mädchen (vgl. Tabelle 6.20). Die unterschiedlichen Leistungssteigerungen können möglicherweise

darauf zurückgeführt werden, dass die im sprachgebundenen Denken Schwachen im Vortest nur sehr schlechte Ergebnisse erzielt hatten (siehe Abbildung 6.8) und sich somit leichter verbessern konnten.

Die Box-Plots der korrespondierenden Merkmalsausprägungen der Variablen *zahlengebunden* und *formallogisches Denken* stimmen fast überein. Die Gruppen der Stärkeren waren etwas erfolgreicher. Die Differenzen ihrer Ergebnisse des Nach- und Vortests waren größer. Wie Abbildung 6.9 zeigt, erzielten die verschiedenen Gruppen auch bezüglich dieser Variablen im Vortest ähnliche Ergebnisse.



Abbildungung 6.9: Häufigkeitsverteilung der Punkte im Vortest getrennt nach den Merkmalen *zahlengebunden* und *formallogisches Denken*

Im Gegensatz zum Merkmal *Geschlecht* zeigte der U-Test von Mann und Whitney für keines der Merkmale *kognitives Gesamtleistungsniveau* ($Z = -0,819, p = 0,413$), *sprachgebunden* ($Z = -2,539, p = 0,011$), *zahlengebunden* ($Z = -1,820, p = 0,069$) und *formallogisches Denken* ($Z = -1,588, p = 0,112$) einen signifikanten Unterschied bezüglich der Differenz der Ergebnisse des Nach- und Vortests in den korrespondierenden Gruppen. Die Nullhypothese, dass sich die Differenzen der Ergebnisse des Nach- und Vortests für die starken und schwachen Gruppen der einzelnen Merkmale nicht signifikant voneinander unterscheiden, kann also beibehalten werden.

Zusammenfassung

Eine signifikant höhere Leistungssteigerung vom Vor- zum Nachtest kann zwischen den Merkmalsausprägungen „stark“ und „schwach“ der Merkmale *kognitives Gesamtleistungsniveau*, *sprachgebunden*, *zahlengebunden* und *formallogisches Denken* nicht nachgewiesen werden. Schülerinnen und Schüler mit einem höheren kognitiven Gesamtleistungsniveau und Stärken im zahlengebundenen und formallogischen Denken erzielten insgesamt ein etwas besseres Ergebnis. Umgekehrt verhält es sich bezüglich des Merkmals *sprachgebunden*. Hier ist die Leistungssteigerung derjenigen mit schlechten Ergebnissen im Verbal-Teil des KFT sehr viel höher.

6.4 Lernerfolg informatischer Grundkonzepte – Unterrichtsmaterialien

Wie bereits in Kapitel 5.3.4 beschrieben kann das Erlernen der informatischen Grundkonzepte nicht umfassend auf Basis der Ergebnisse eines Leistungstests erfasst werden. Aus diesem Grund werden für alle informatischen Grundkonzepte detailliert Arbeitsblätter, Tests und Programme evaluiert. In einer sich anschließenden Analyse wird paarweise der Erfolg folgender Gruppen überprüft und miteinander verglichen:

- Jungen ($N = 13$) und Mädchen ($N = 17$)
- Kinder mit starken ($N = 12$) und schwachen ($N = 12$) Leistungen im KFT (*kognitives Gesamtleistungsniveau*)
- Kinder mit starken ($N = 12$) und schwachen ($N = 12$) Leistungen im Verbal-Teil des KFT (*sprachgebundenes Denken*)
- Kinder mit starken ($N = 12$) und schwachen ($N = 12$) Leistungen im Quantitativ-Teil des KFT (*zahlengebundenes Denken*)
- Kinder mit starken ($N = 12$) und schwachen ($N = 12$) Leistungen im Nonverbal-Figural-Teil des KFT (*formallogisches Denken*)

Lag der Anteil derjenigen, die eine Aufgabe korrekt lösten, in den Gruppen korrespondierender Merkmalsausprägungen mehr als 30 Prozent auseinander, wird dies dargestellt, um die in Kapitel 6.3 vorgestellten Ergebnisse aus dem Leistungstest zu den informatischen Grundkonzepten zu überprüfen. Falls nicht anders angegeben, handelt es sich bei den Stichproben immer um die 30 Schülerinnen und Schüler der Klasse ($N = 30$).

6.4.1 Wirkprinzipien von Informatiksystemen

Algorithmisierung

Nach Schubert und Schwill (2004, S. 5) ist ein *Algorithmus* ein mit formalen Mitteln beschreibbares, mechanisch nachvollziehbares Verfahren zur Lösung einer Klasse von Problemen.

Im Musikunterricht wurde eine textuelle algorithmische Beschreibung verwendet, um Dur- und Moll-Tonleitern einzuführen (vgl. Kapitel 4.1, S. 67). Analog entwickelten die Schülerinnen und Schüler selbst eine Zahlen- und eine Buchstabenreihe als Rätsel. Bei den Zahlenreihen waren 10 der 24 Lösungen völlig korrekt. Sechs weitere enthielten ungenaue Formulierungen in Form von relativen Beschreibungen wie zum Beispiel „die nächsten Zahlen sind alle immer eins größer“, für die keine Bezugspunkte festgelegt worden waren. Die Rätsel konnten jedoch trotzdem gelöst werden. In Tabelle 6.21 ist dargestellt, auf welche Fehlerkategorien sich die acht fehlerhaften Beschreibungen verteilen.

Fehlerkategorien	Anzahl
Keine komplette Beschreibung	5
Inkonsistenzen in der Beschreibung	2
Chaotische Beschreibung	1

Tabelle 6.21: Fehlerkategorien bei der Formulierung der Zahlenreihe als Algorithmus ($N = 8$)

In einer unvollständigen Formulierung war zum Beispiel nicht angegeben, wie viele Elemente die Zahlenreihe enthielt. In einer anderen, inkonsistenten Darstellung entstand auf Basis der ersten sechs Anweisungen die Lösung in Abbildung 6.10.

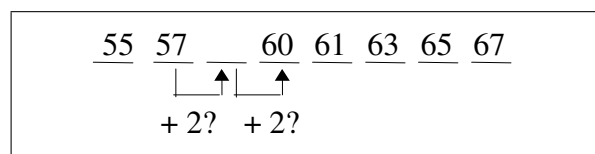


Abbildung 6.10: Zahlenreihe eines Schülers

Die letzte Anweisung lautete „Der Rest ist immer plus 2, außer denen, die schon dort stehen“. Es existierte also keine Lösungsmöglichkeit für das Rätsel. Formulierungen, die ein mechanisches Vorgehen ermöglichen, waren, wie Abbildung 6.11 zeigt, vor allem für die im sprachgebundenen Denken starken Kinder noch sehr schwierig.

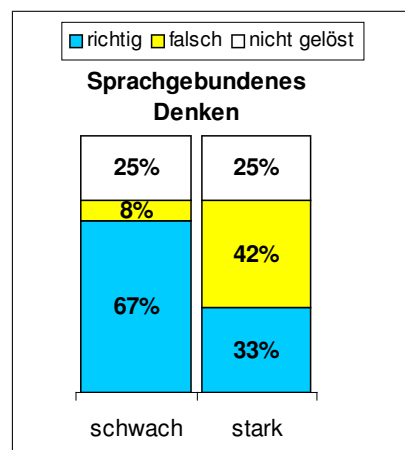


Abbildung 6.11: Erfolg bei der algorithmischen Formulierung der Zahlenreihe

Das Ergebnis veränderte sich auch in der zweiten Übung der Buchstabenreihen nicht. Hier waren 10 der 23 Algorithmen korrekt, fünf enthielten ungenaue Formulierungen und die Fehlertypen blieben gleich. Wie in Abbildung 6.12 dargestellt, gab es bezüglich fast aller kognitiven Fähigkeiten und des Geschlechts Unterschiede bei der Formulierung eines exakten Verfahrens zur Lösung des Buchstabenrätsels.

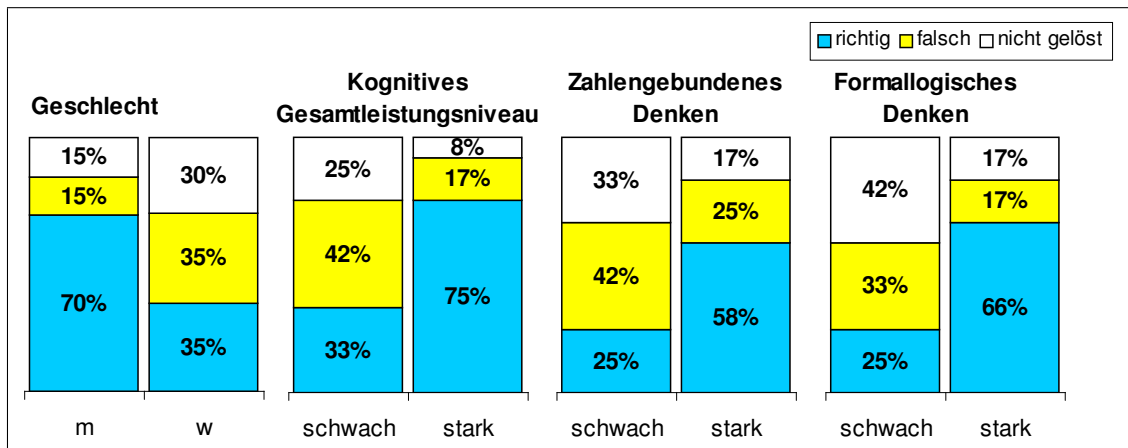


Abbildung 6.12: Erfolg bei der algorithmischen Formulierung der Buchstabenreihe

Die in Kapitel 4.1 (S. 74 ff.) dargestellte Umwandlung römischer Zahlen in Dezimalzahlen wurde in Textform beschrieben. Die Ergebnisse der Abarbeitung des Algorithmus sind in Tabelle 6.22 abgebildet.

Umwandlung von MCMLXIX	Anzahl
korrekt	24
falsch	4
Umwandlung von MCDXIV	
korrekt	17
falsch	5
nicht gelöst	6
Umwandlung von CDLXIX	
korrekt	24
falsch	4

Tabelle 6.22: Ergebnis der Umwandlung römischer Zahlen in Dezimalzahlen ($N = 28$)

Der Algorithmus sah vor, dass die Subtrahenden rechts und die Summanden links aufsteigend angeordnet wurden. Trotzdem sortierten 28 Prozent der 28 Schülerinnen und Schüler die Summanden und Subtrahenden nicht entsprechend und 7 Prozent vermischten sie sogar. Die Ungenauigkeiten führten nicht zwangsläufig zu Fehlern. Dies ist vermutlich darauf zurückzuführen, dass die Schülerinnen und Schüler ein intuitives Verständnis des Algorithmus hatten. Aus diesem Grund lasen sie die Anweisungen nicht genau und ignorierten formale Inhalte wie „Streiche das erste Zahlzeichen aus der römischen Zahl“ (Kapitel 4.1, S. 75). Diese Inhalte sind zwar für ein mechanisches Abarbeiten der Vorschriften notwendig, für die Berechnung jedoch nicht relevant.

Die Dominanz der Schülerinnen und Schüler mit einem höheren kognitiven Gesamtleistungsniveau und besseren Ergebnissen im KFT im Bereich des zahlengebundenen und formallogischen Denkens bei der Umwandlung der zweiten römischen Zahl ist sehr deutlich. Wie schon bei der Beschreibung der Buchstabenreihe steht der Erfolg in keinem Zusammenhang mit dem sprachgebundenen Denken.

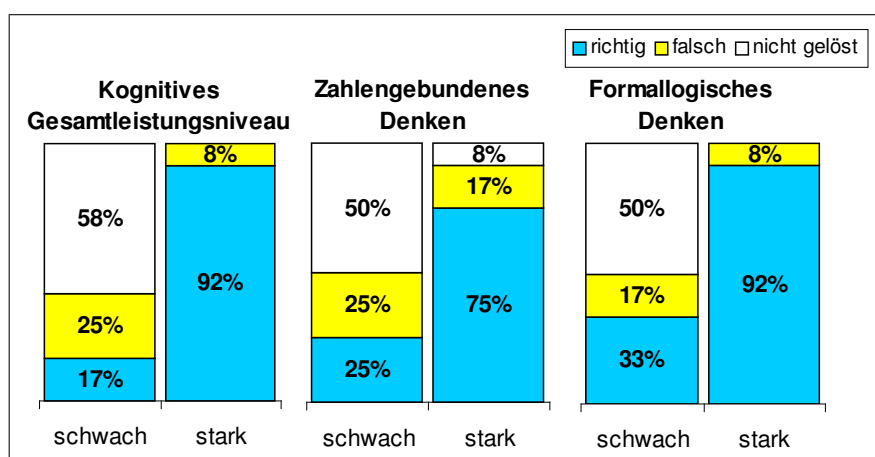


Abbildung 6.13: Erfolg bei der Umwandlung der römischen Zahl MCDXIV in das Dezimalsystem

Die Fähigkeit, algorithmische Beschreibungen exakt abzuarbeiten, scheint unabhängig von der Darstellungsform zu sein. Bei der Dekodierung einer Geheimschrift mit Hilfe eines Programmablaufplans (Anhang G, S. 247) waren die Ergebnisse kaum besser. 19 von 30 Schülerinnen und Schülern entschlüsselten unabhängig von ihren Fähigkeiten alle vier Wörter richtig. Auch hier wurden wieder zahlreiche Zwischenschritte, wie zum Beispiel das abschließende Vertauschen des ersten und letzten Buchstabens und das Schreiben des Ergebnisses in Schreibschrift, nicht dargestellt, sondern einfach das dekodierte Wort notiert (vgl. Tabelle 6.23). Ein Großteil der Kinder ist also nicht bereit, die Hilfestellungen, die durch die genaue Beschreibung der vielen Zwischenschritte angeboten werden, anzunehmen. Der Aufwand der konsequenten Abarbeitung der einzelnen Schritte ist ihnen zu hoch.

Fehlende Zwischenschritte	Anzahl
mindestens ein Hilfsword fehlt	6
mindestens in einem Wort ist die Ersetzung nicht sichtbar	15
mindestens einmal ist die Vertauschung der Buchstaben nicht dargestellt	25
mindestens ein Wort in Schreibschrift fehlt	11

Tabelle 6.23: Fehlende Zwischenschritte bei der Dekodierung von vier Wörtern mit Hilfe eines Programmablaufplans ($N = 30$)

Automatisierung

Unter *Automatisierung* wird in dieser Arbeit die Zusammenfassung wiederkehrender Abläufe verstanden, so dass sie von Routinetätigkeiten entlastet.

Mit der Automatisierung beschäftigten sich die Schülerinnen und Schüler schon zu Beginn des Unterrichts beim Zeichnen ebener Figuren. Befehle wurden einzeln eingegeben, im WIEDERHOLE-Befehl zusammengefasst und in Form neuer Befehle abstrahiert. Bei der Erstellung einer Drehfigur aus geschachtelten Quadraten (vgl. Kapitel 4.1, S. 68 f.) bot es sich an, zunächst einen Befehl `Quadrat` mit Eingabeparameter zu formulieren, um mit ihm das Grundelement der Drehfigur in einem eigenen Befehl `Quadratfigur` zu erstellen. Auf 18 der 26 abgegebenen Arbeitsblätter wurden `Quadratfigur`-Befehle notiert. Zwölf davon waren korrekt. Die Hauptfehlerursache der sechs anderen war die Zusammenfassung der Aufrufe in eckigen Klammern. Vermutlich wurden die eckigen Klammern analog zum WIEDERHOLE-Befehl gesetzt.

PR Quadratfigur

[Quadrat 20 Quadrat 40 Quadrat 60 Quadrat 80 Quadrat 100]

ENDE

Die Hälfte der acht fehlenden `Quadratfigur`-Befehle ist auf die Kinder zurückzuführen, die folgende Lösung notierten:

Quadrat 20 Quadrat 40 Quadrat 60 Quadrat 80 Quadrat 100

Den Aspekt der Automatisierung durch Zusammenfassung einer Befehlsfolge in einem neuen Befehl, der mehrfach verwendet wird, beachteten sie nicht. Zum Zeichnen der gedrehten Figur mussten sie die Befehlsfolge mehrfach eingeben.

Wie Abbildung 6.14 zeigt, waren vor allem die Jungen und die im sprachgebundenen Denken schwächeren und zahlengebundenen Denken stärkeren Kinder erfolgreicher.

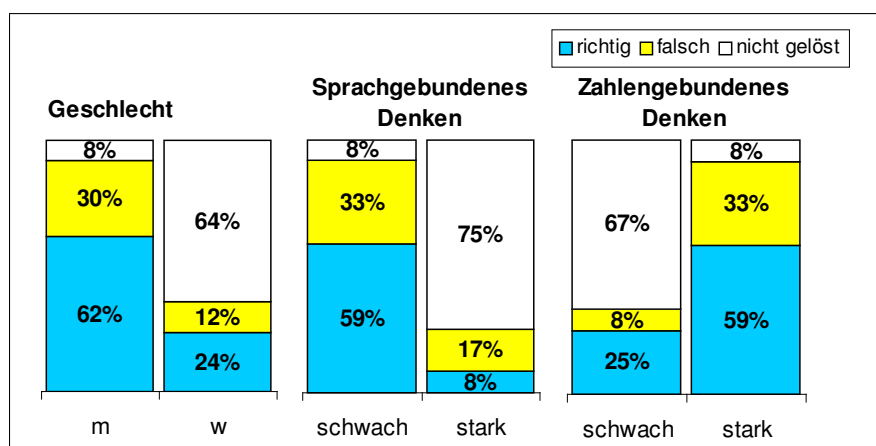


Abbildung 6.14: Erfolg bei der Erstellung des Befehls `Quadratfigur`

In der jeweils anderen Gruppe wurde von einem sehr hohen Anteil der Schülerinnen und Schüler überhaupt kein `Quadratfigur`-Befehl erstellt.

Nachdem die Kinder beim Zeichnen der Drehfigur nur sehr wenig Bereitschaft zeigten, Schwierigkeiten selbstständig zu lösen und neue Figuren auszuprobieren, verhielten sie sich bei der Erstellung der Fadenbilder (siehe Kapitel 4.1, S. 80) völlig anders. Sie experimentierten von Beginn an mit den Bewegungen der beiden Igel und fanden schnell heraus, wie interessante Bilder entstehen (vgl. Abbildung 6.15).

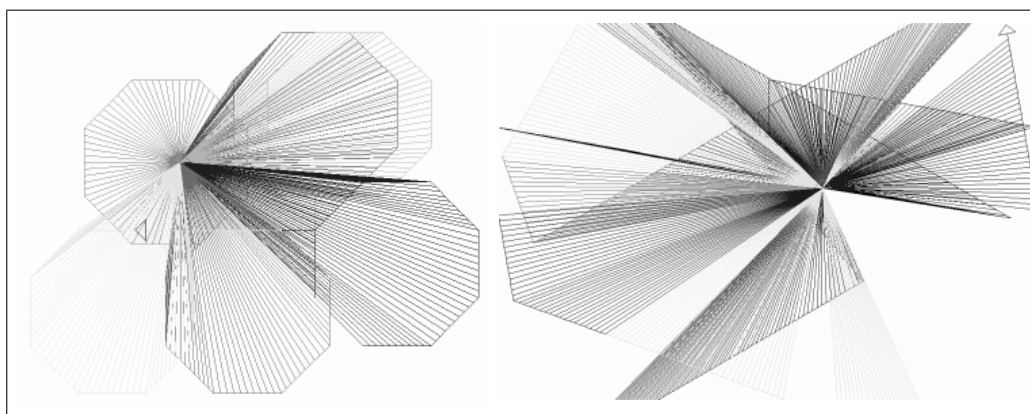


Abbildung 6.15: Automatisierte Fadenbilder

Den Prozess des Zeichnens automatisierten sie dadurch, dass sie die kurzen Bewegungen und den Wurf des Fadens in `WH`-Befehlen³ kapselten.

```
WH 8 [ WH 10 [ VORWÄRTS 10 Faden_zum_zweiten ] RE 45 ]
```

Mit dem inneren `WIEDERHOLE`-Befehl wurde jeweils eine Seite des Achtecks gezeichnet.

Bei der Arbeit an den Fadenbildern hatten alle Kinder Erfolgserlebnisse und lernten das Konzept der Automatisierung als Arbeitserleichterung schätzen. Im Gegensatz zu anderen Unterrichtseinheiten versuchten sie ihre ersten Lösungen durch Veränderung der Schrittweiten und Drehwinkel und den Einsatz von Farben zu verbessern.

In der Rittergeschichte bauten die Schülerinnen und Schüler zahlreiche Bewegungen in Form von automatisierten Animationen (vgl. Kapitel 4.2, S. 96 f.) in ihre Projekte ein. Es bewegten sich Wachsoldaten vor einer Ritterburg, ein Drache rannte eingesperrt in einem Käfig umher und Vögel flogen am Himmel. Wie schon im Test zu den informatischen Grundkonzepten wurden die Animationen, die eine exakte formale Beschreibung der Bewegungsabläufe erforderten, vor allem von den Jungen, den im sprachgebundenen Denken schwachen und den im zahlengebundenen und formallogischen Denken starken Kindern erstellt.

³`WH` ist in Logo die abkürzende Schreibweise für `WIEDERHOLE`

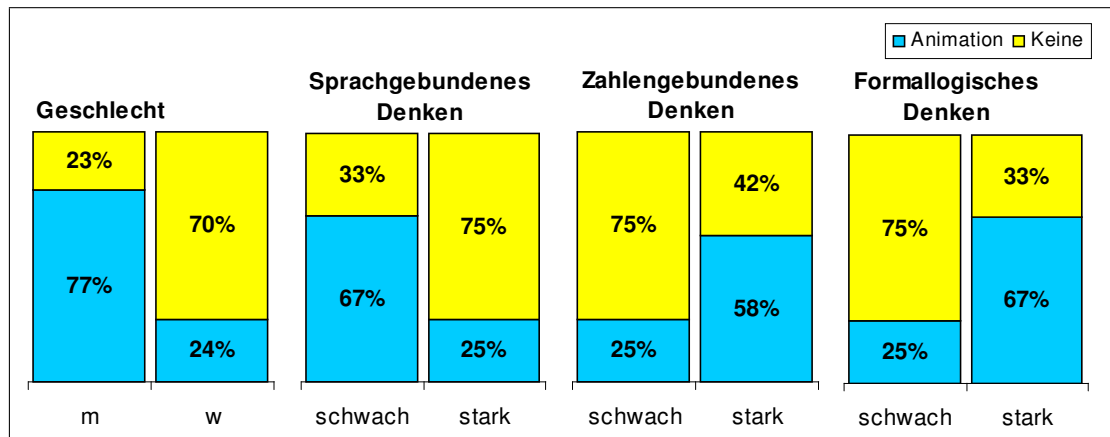


Abbildung 6.16: Erfolg bei der Erstellung automatisierter Animationen in der Rittergeschichte

Codierung

Codes sind Abbildungsvorschriften, die jedem Zeichen eines Zeichenvorrats (Urbildmenge) eindeutig ein Zeichen oder eine Zeichenfolge aus einem möglicherweise anderen Zeichenvorrat (Bildmenge) zuordnen (Claus und Schwill, 2003).

Codes traten im Laufe des Schuljahrs häufiger auf. In Musik waren zum Beispiel Töne zunächst als Zahlen, in Deutsch Wörter in Geheimschrift und in Mathematik Dezimalzahlen als römische Zahlen codiert. Selbstständig entwickelten die Schülerinnen und Schüler nur im Test zu den informatischen Grundkonzepten einen Code für Geburtsdaten (vgl. Anhang B, S. 219). Im Vortest entwarfen 5 und im Nachtest 13 Schülerinnen und Schüler einen korrekten Code. Fehlte die Abbildungsvorschrift für den Punkt, so spielte das bei der Bewertung keine Rolle. Alle Kinder, die die Aufgabe zu Beginn des Schuljahrs lösten, waren auch im Nachtest erfolgreich.

Die von den Schülerinnen und Schülern für die Codes gewählten Urbildmengen können in vier Kategorien aufgeteilt werden:

- Zahlen von 0 bis 9 und Punkt
- Zahlen von 0 bis 9
- Nur die für die Anwendung notwendigen Zahlen (unvollständig)
- Zu viele Zahlen (keine eindeutige Codierung möglich).

Der Schüler, der die Lösung in Abbildung 6.17 ganz links erstellte, codierte alle Zahlen von 0 bis 9 und den Punkt. Die Lösung der Schülerin in der Mitte berücksichtigte nur die Zahlen, die für die Daten „20.03.1993“ und „30.12.1786“ notwendig waren. Aus diesem Grund fehlte die Abbildungsvorschrift für die Ziffer „4“. Ganz rechts wurden Abbildungsvorschriften für die Zahlen von 0 bis 12 notiert. Der Monat Dezember wurde mit „SA“ codiert. Möglich wäre auch „J“ gewesen.

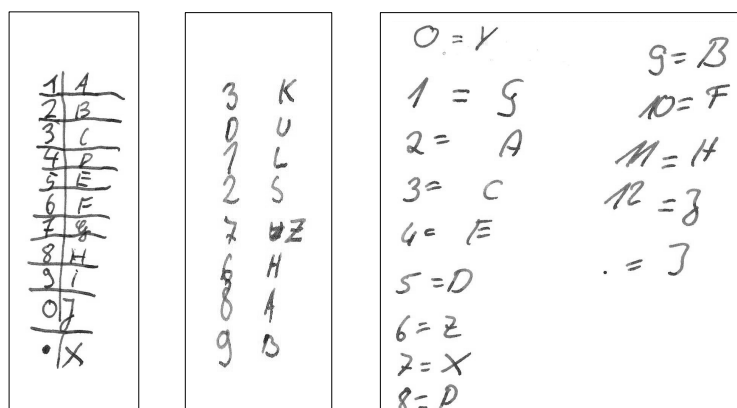


Abbildung 6.17: Codes der Schülerinnen und Schüler für Daten

Abbildung 6.18 zeigt die Verteilung der von den Schülerinnen und Schülern entwickelten Urbildmengen im Vortest (18 Lösungsversuche) und im Nachtest (27 Lösungsversuche).

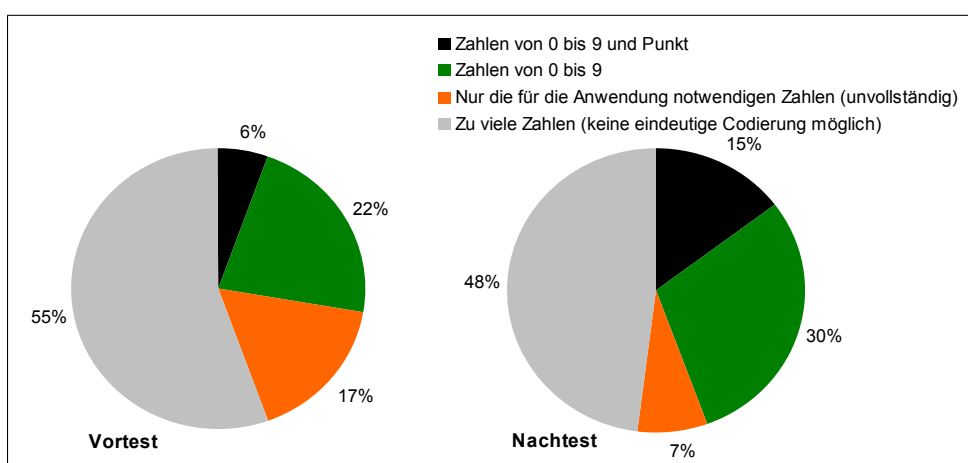


Abbildung 6.18: Codetypen für Geburtsdaten

Interessant sind die verschiedenen Notationsformen des Codes. Die Abbildungsvorschriften wurden zum Beispiel mit Hilfe von Gleichheitszeichen (Abbildung 6.17 rechts), mit Bindestrichen, durch vertikale und horizontale (Abbildung 6.17 Mitte) Gegenüberstellungen oder innerhalb einer Tabelle (Abbildung 6.17 links) in aufeinander folgenden Zellen in einer Zeile dargestellt. Es erfolgte immer eine direkte Zuordnung eines Elementes aus der Urbildmenge zu einem Element in der Bildmenge. Lediglich eine Schülerin definierte im Nachtest die Urbild- und Bildmenge getrennt für Tage, Monate und Jahre. Sie gab nicht für jedes Element einzeln eine Abbildungsvorschrift an. Diese war vielmehr in den Worten *von* und *ab* implizit enthalten. Für die Jahreszahl wurde sie durch die Verbindung der einzelnen Ziffern durch Linien verdeutlicht (vgl. Abbildung 6.19).

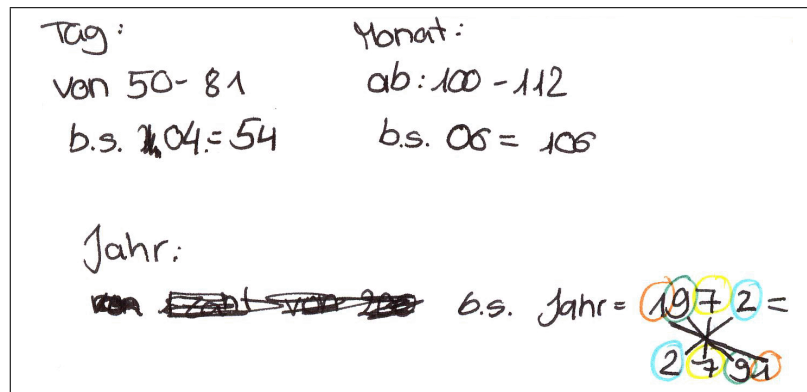


Abbildung 6.19: Abbildungsvorschrift eines Codes

Zur Darstellung der Abbildung der Elemente der Urbildmenge auf die Elemente der Bildmenge wurden die Elemente der Urbildmenge typischerweise links und die der Bildmenge rechts angeordnet. Erstaunlicherweise war dies bei 33 Prozent der Fälle im Vortest und 25 Prozent im Nachtest nicht der Fall. Da in den folgenden beiden Aufgaben nur Geburtsdaten codiert und nicht dekodiert wurden, konnte nicht geklärt werden, ob bei einer Vermischung der Aufgaben eine höhere Fehlerquote durch diese unübliche Anordnung auftreten würde. Dies ist vor allem bei identischen Urbild- und Bildmengen zu erwarten.

Abbildung 6.20 zeigt, dass bei der Erstellung der Codes zu den Geburtsdaten die im sprachlichen Denken stärkeren Kinder wesentlich erfolgreicher waren. Dies widerspricht den in Kapitel 6.3.2 beschriebenen Ergebnissen der Tests zu den informatischen Grundkonzepten.

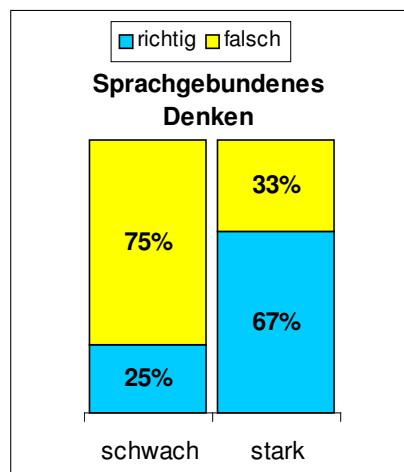


Abbildung 6.20: Erfolg beim Entwurf des Codes für Daten

6.4.2 Informatische Modellierung

Datenstrukturen

Mit Hilfe von *Datenstrukturen* werden Wertebereiche aus elementaren, d.h. nicht weiter zerlegbaren Wertebereichen mit Hilfe von Konstruktoren aufgebaut (Claus und Schwill, 2003). Die Datenstrukturen dienen dann zur Verwaltung und Verknüpfung von Daten, so dass auf diese zugegriffen werden kann.

Im Unterricht entwickelten die Schülerinnen und Schüler die Datenstrukturen nicht mit Hilfe von Konstruktoren, sondern implementierten sie direkt als Logo-Listen. Im zweiten Test während des Schuljahrs schrieben sie Logo-Funktionen, mit denen die Sätze

- Ben can play football.
- Sanjay can sing a song.
- Jenny can draw a picture.

automatisch generiert wurden (vgl. Anhang G, S. 245). Wichtig war dabei, dass *a song* und *a picture* in eckigen Klammern geschachtelt wurden.

```
[ football [a song] [a picture] ]
```

14 der 25 Lösungen enthielten Fehler. Die Datenstrukturen waren entweder aus anderen Sätzen aufgebaut ([[der Brief] [die Leiter]]), die Objekte auf die Wörter *song* und *picture* beschränkt ([football song picture]) oder es fehlten einfach die eckigen Klammern ([football a song a picture]). Die Datenstruktur an sich war nur im letzten Fall fehlerhaft. Annähernd gleich verteilten sich die korrekten Lösungen auf die korrespondierenden Gruppen der Schülerinnen und Schüler.

Modellierung

Ein *Modell* ist nach Baumann (1996, S. 161) eine vereinfachte struktur- und verhaltenstreue Beschreibung eines realen Systems. Bei der *Modellierung* handelt es sich ganz allgemein gesprochen um den Entwurf eines Modells. Dies schließt natürlich eine Bearbeitung und Erweiterung ein.

Beim Autospiegel (vgl. Kapitel 4.3, S. 97 ff.) wurden sowohl das Modell der Benutzerschnittstelle als auch der interne Aufbau des Programms erweitert. Nur zehn Schülerpaare speicherten ihre Lösung ab. Diese enthielten zwei bis acht neue Spiel-Modi, für die an der Oberfläche weitere Schaltflächen eingefügt worden waren. Ihre Beschriftung ließ in fast allen Fällen einen Rückschluss auf den Schwierigkeitsgrad des Modus zu. Außerdem waren sie bezüglich der Schwierigkeitsstufen fallend oder aufsteigend angeordnet. Sechs Paare achteten auch auf die äußere Gestalt der Schaltflächen. Es zeigte sich, dass die Sensibilität der Schülerinnen und Schüler für die Benutzerfreundlichkeit und den ästhetischen Gesamteindruck ihres Produkts sehr hoch war. So wurde das Modell der Benutzerschnittstelle stimmig erweitert.

Auch bei der Benennung neuer Befehle für verschiedene Schwierigkeitsstufen war in allen Lösungen eine Systematik zu erkennen. Das prinzipielle Vorgehen wurde durch das vorherige Codestudium mit Hilfe von Arbeitsblättern vorbereitet, so dass sich die Kinder an der konsistenten Nomenklatur der verschiedenen Module, die zum Teil in Abbildung 6.21 dargestellt sind, orientieren konnten.

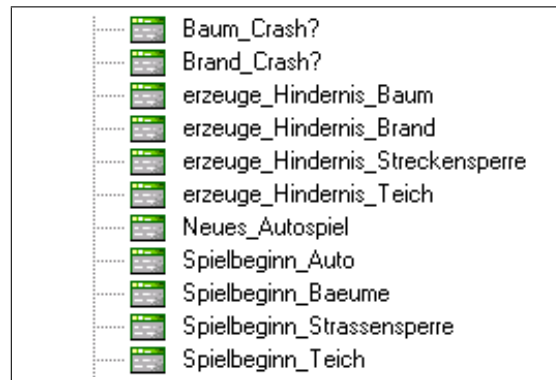


Abbildung 6.21: Nomenklatur der Befehle im Autospiel

Fünf Paare wählten Bezeichnungen, in denen der Spielmodus sichtbar wurde. Sie nannten sie zum Beispiel **Anfänger**, **Fortgeschrittener**, **Profi** oder **10kmh**, **50kmh**, **100kmh**. Die Verhältnisse der Geschwindigkeiten im letzten Beispiel spiegelten sich allerdings nicht im Programmcode wider. Ein Paar nutzte für die Benennung seiner Befehle Abkürzungen wie **n** (normal), **p** (Profi) oder **s** (super schnell). Drei weitere Lösungen nahmen den vorgegebenen Befehlsnamen **starte_Autofahrt** und hängten entweder Nummern an (**starte_Autofahrt_2**) oder vergaben Namen wie **Starte_Aut**, **Starte.Auto**, **Starte_Autof** etc. Das interne Modell des Spiels wurde durch die Erweiterung der Funktionalität konsistent verändert.

Objektorientierung

Die *Objektorientierung* bezeichnet ein Paradigma für die Analyse und den Entwurf von Systemen, die aus gleichberechtigten Objekten aufgebaut sind. Grundlegende Prinzipien der Objektorientierung sind u.a. *Objekte*, *Klassen*, *Attribute*, *Methoden*, *Nachrichten*, *Vererbung* und *Identität*.

Am Ende des Schuljahrs beherrschten die Schülerinnen und Schüler auf der Ebene der Programmierung einige objektorientierte Konzepte. Sie kannten Klassen des Imagine Logo Systems (z.B. **TURTLE**, **TEXTBOX**) mit Attributen und Methoden, benannten die Objekte der Klassen (Identität) und riefen entsprechende Methoden auf. Sie bewegten in der Rittergeschichte (**FOREVER [knight'FD 20 WAIT 100 knight'RT 786]**) oder der Bildergeschichte (**FOREVER [Jim'FD 20 WAIT 100]**) Figuren, veränderten die Attributwerte der Textfelder im Rätsel der Rittergeschichte (**question1'SETBGCOLOUR 12**)

oder überprüfen im Labyrinthspiel die Berührung der Maus mit der Wand (`IF EQUAL? mouse'DOTCOLOUR "red"`).

6.4.3 Arbeitstechniken

Modularisierung und Hierarchisierung

Ein *Modul* ist ein Teilsystem mit klaren Grenzen (Schnittstelle) zu seiner Umgebung. Die *Modularisierung* ist der Prozess der Zerlegung eines Systems in Module (Baumann, 1996, S. 261). Im Gegensatz dazu handelt es sich bei der *Hierarchisierung* um einen Prozess der Zusammensetzung eines komplexen Systems aus Modulen.

Die informatischen Grundkonzepte der Modularisierung und Hierarchisierung wurden wie auch die Objektorientierung in sehr vielen Unterrichtseinheiten angewendet. In einigen Situationen wurden vorgegebene Module genutzt, um eine konkrete Aufgabenstellung zu lösen. Für die Weihnachtskarte in Abbildung 6.22 verwendete der Schüler zum Beispiel das Modul *Kerze* zur Erzeugung eines Rahmens um die Karte und *Dreieck* zur Gestaltung des Sterns in der Mitte.

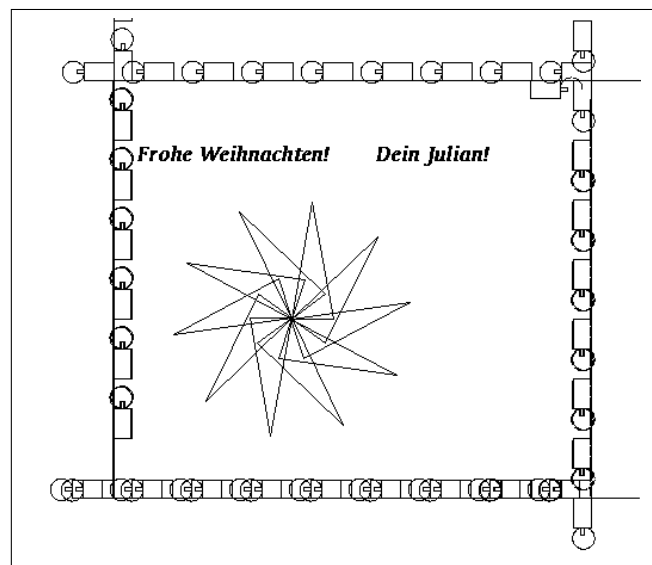


Abbildung 6.22: Weihnachtskarte, zusammengesetzt aus den Modulen *Kerze* und *Dreieck*

Beim Entwurf der Satz- und Fragegeneratoren mussten die Schülerinnen und Schüler die Problemstellungen selbst in Teilprobleme zerlegen und für diese eigenständige Lösungen in Form von Funktionen für die einzelnen Satzteile implementieren. Diese setzten sie dann mit der Funktion `SATZ` bzw. `SENTENCE` zu Generatoren zusammen.

Die Fähigkeit der Kinder, die Modularisierung und Hierarchisierung anzuwenden, werden im folgenden Abschnitt *Abstraktion* und in den Kapiteln 6.4.4 und 6.4.5 bei der Analyse der informatischen Grundkonzepte *Syntaxdiagramm*, *Grammatik* und *Funktion* ausführlich beschrieben.

Abstraktion

Die *Abstraktion* ist ein Verallgemeinerungsprozess.

In zahlreichen Unterrichtsbeispielen abstrahierten die Schülerinnen und Schüler Satzstrukturen aus Beispielsätzen, beschrieben diese mit Maschinen und Syntaxdiagrammen und implementierten sie in Logo als Funktionen.

Im ersten Test in Aufgabe 4 leiteten sie aus drei Fragen der Art „Is Nick the brother of Debbie?“ (vgl. Anhang G, S. 241) eine zusammengesetzte Maschine ab. Sie abstrahierten die Namen *Nick*, *Ben* und *Debbie* in einer Maschine *name*. 21 Kinder lösten die Aufgabe richtig. Sie schrieben nicht eine der Fragen direkt an die Eingabetrichter der Funktion **SENTENCE**, sondern zeichneten die Maschine *name*. Abbildung 6.23 zeigt die Lösung eines Schülers. Der Datenfluss von oben nach unten durch die Eingabe- und Ausgabetrichter der Maschinen ist deutlich zu erkennen.

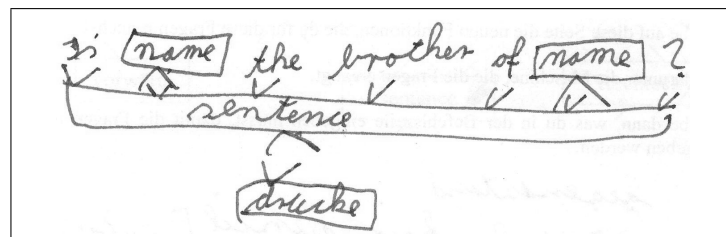


Abbildung 6.23: Zusammengesetzte Maschinen – Englische „Is“-Fragen

14 Lösungen enthielten auch den unteren Ausgabetrichter der Funktion. Sieben Lösungen waren falsch und nur zwei Kinder bearbeiteten die Aufgabe überhaupt nicht. Im zweiten Test drei Monate später, in dem Satzstrukturen mit Syntaxdiagrammen beschrieben wurden, war keine Leistungssteigerung zu erkennen (vgl. Kapitel 6.4.4, S. 168 ff.). Wie schon bei der Entwicklung der Datenstruktur der geschachtelten Liste (siehe Kapitel 6.4.2, S. 159) verteilten sich die korrekten zusammengesetzten Maschinen etwa gleich auf die verschiedenen Gruppen der Schülerinnen und Schüler.

Anders sahen die Ergebnisse beim Lesen der zusammengesetzten Maschinen aus. Im ersten Test in Aufgabe 3 (vgl. Anhang G, S. 240) mussten zwei Fragen der Art „Is Ben from Asthon?“ aus der vorgegebenen zusammengesetzten Maschine abgeleitet werden. Hier waren die Kinder erfolgreicher, die im KFT insgesamt und im Bereich des formallogischen Denkens besser waren (vgl. Abbildung 6.24). Insgesamt konnten 22 Schülerinnen und Schüler die Fragen richtig formulieren.

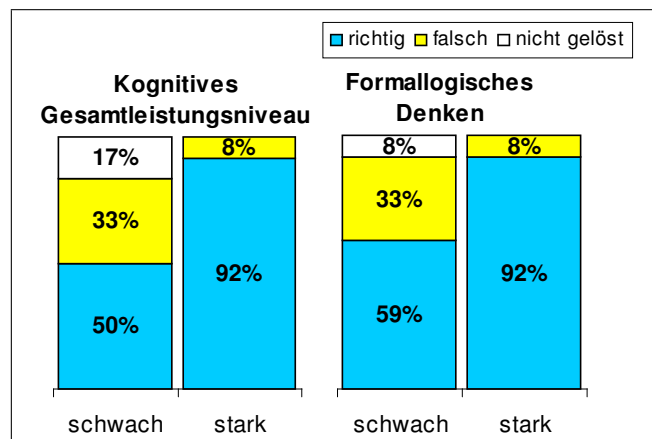


Abbildung 6.24: Erfolg beim Lesen der zusammengesetzten Maschinen englischer „Is“-Fragen

Iteration

Unter Iteration wird die Wiederholung von Anweisungen oder Anweisungsfolgen verstanden.

In einer Unterrichtseinheit zu den ebenen Figuren mussten die Schülerinnen und Schüler das Zeichnen eines Rechtecks iterativ beschreiben (vgl. Kapitel 4.1, S. 66). 18 Kinder lösten die Aufgabe richtig, acht Beschreibungen waren falsch und vier Arbeitsblätter wurden nicht bearbeitet. Die fehlerhaften Aufgaben verteilten sich auf die in Tabelle 6.24 dargestellten vier Kategorien.

Fehlerkategorien	Anzahl
kein WIEDERHOLE-Befehl vw 100 re 90 vw 50 re 90 vw 100 re 90 vw 50 re 90	3
vierfache Wiederholung wh 4 [vw 100 re 90 vw 50 re 90]	2
Igel wird nur einmal gedreht wh 2 [vw 100 re 9 vw 50]	1
keine Struktur zu erkennen	2

Tabelle 6.24: Fehlerkategorien bei der iterativen Beschreibung des Programmcodes eines Rechtecks

Zwei wiederholten die Folge der Anweisungen viermal wie beim Quadrat. In drei Lösungen fehlte der WIEDERHOLE-Befehl ganz. Einmal wurde vergessen, dass sich der Igel nach jeder Bewegung drehen muss. In der Befehlsfolge vw 100 re 9 vw 50 wurde nur beachtet, dass ein Rechteck zwei verschieden lange Seiten hat. In den anderen fehlerhaften Lösungen war überhaupt keine Struktur zu erkennen.

Abbildung 6.25 zeigt, dass vor allem wieder die Kinder mit einem hohen kognitiven Gesamtleistungsniveau und Stärken im formallogischen Denken den Abstraktionsschritt zu einer iterativen Beschreibung vollziehen konnten.

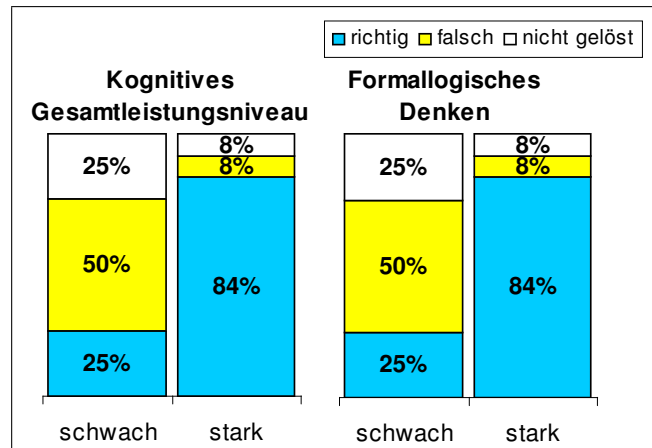


Abbildung 6.25: Erfolg bei der iterativen Beschreibung eines Rechtecks

Rekursion

Rekursion ist die Definition einer Funktion oder eines Verfahrens durch sich selbst (Claus und Schwill, 2003).

Beim Zeichnen von Figuren auf Basis von rekursiven Beschreibungen steigerten sich die Schülerinnen und Schüler im Test zu den informatischen Grundkonzepten (vgl. Anhang B, S. 204) vom Schuljahresbeginn zum Schuljahresende erheblich. Im Vortest konnten lediglich 4, im Nachtest immerhin 20 Kinder eine rekursive Beschreibung in eine Zeichnung umsetzen.

Auch im Unterricht übten die Schülerinnen und Schüler das Zeichnen von rekursiven Figuren (vgl. Kapitel 4.3). Die Auswertung der Arbeitsblätter zeigte, dass fast alle Figuren während der eigenen Arbeit oder der Besprechung im Klassenverband korrigiert wurden. Trotzdem enthielten viele Arbeitsblätter noch Fehler. Dies wurde vor allem an der zweiten Figur deutlich, die nach folgender Vorschrift gezeichnet wurde:

Aufgabe:

„Kreisrekeln“ bedeutet, in einem Kreis in der linken Hälfte einen halb so großen Kreis zu zeichnen und diesen kleineren neuen Kreis dann zu „Kreisrekeln“.

Die Fehlertypen sind in Tabelle 6.25 und entsprechende Beispiele in Abbildung 6.26 abgebildet. Im ersten Fall wurden in der rechten Hälfte Ovale gezeichnet. Im zweiten Bild befindet sich nur der äußerste Kreis in der linken Hälfte seines übergeordneten Kreises. Die im Rekursionsaufruf konstruierten Kreise wurden dann jeweils in die Mitte

KatNr	Fehlerkategorien	Anz.
1	Kreise in der rechten Hälfte	4
2	Kreise aus dem Rekursionsaufruf in der Mitte und nicht in der linken Hälfte	4
3	Ovale statt Kreise	9
4	Keine Halbierung des Kreisdurchmessers	16
5	Kreise in der Mitte	1

Tabelle 6.25: Fehlerkategorien beim Zeichnen der rekursiven Kreisfigur ($N = 29$; Mehrfachnennungen möglich)

gezeichnet. Ganz rechts wurden die Kreise schon von Beginn an in der Mitte platziert. In allen drei Fällen fehlte außerdem die Halbierung des Kreisdurchmessers.

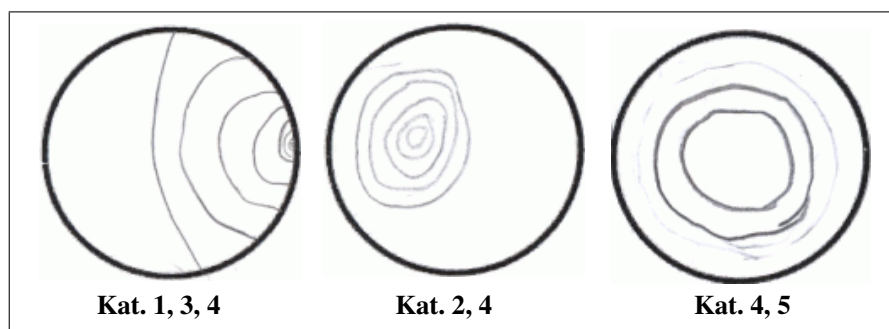


Abbildung 6.26: Fehlerhafte Darstellungen rekursiver Kreisfiguren

Beim abschließenden selbstständigen Zeichnen einer Figur, die „nach dem gleichen Prinzip aufgebaut“ war, wurden vor allem von den Jungen (siehe Abbildung 6.27) 18 Lösungen entwickelt. Eine davon wiederholte nur die Kreisfigur und in einer weiteren wurden Kreuze und Rechtecke vermischt.

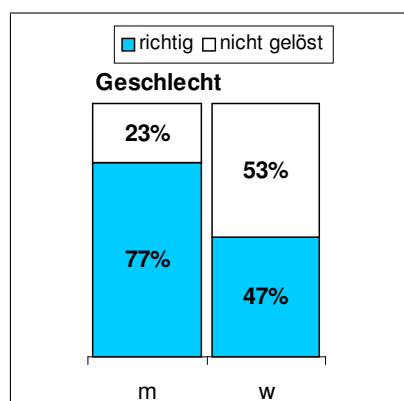


Abbildung 6.27: Erfolg bei der selbstständigen Entwicklung einer rekursiven Figur

Anders als bei der Umsetzung der rekursiven Beschreibungen in Figuren, in der die Halbierung der Kreisdurchmesser nicht beachtet wurde, sind in der Dreiecksfigur links noch die Mittelsenkrechten der verschiedenen Dreiecke erkennbar. Der Schüler benutzte diese als Hilfslinien, um eine exakte Lösung zu konstruieren. Nach Fertigstellung der Zeichnung radierte er sie wieder aus.

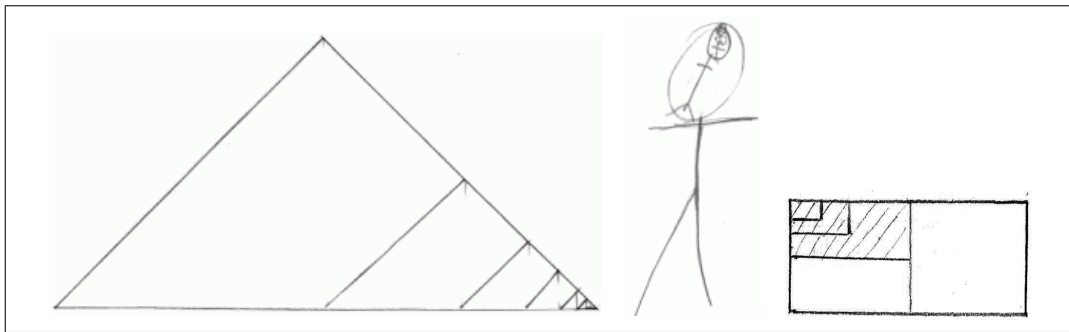


Abbildung 6.28: Rekursive Figuren der Schülerinnen und Schüler

Neun Kinder beschrieben zusätzlich zu ihrer rekursiven Figur noch den Zeichenprozess. Allerdings gelang dies nur sehr wenigen rekursiv. Das Strichmännchen in Abbildung 6.28 in der Mitte wurde von dem Schüler zum Beispiel folgendermaßen iterativ beschrieben:

„Strichmännrekeln: Malen sie ein Strichmännchen. Malen sie in den Kopf ein Strichmännchen. Und jetzt wieder das immer wiederholen“

Eine korrekte rekursive Beschreibung lieferte eine Schülerin für die Figur rechts:

„*Rechtforekeln* bedeutet im linken oberen Teil des Rechtecks ein neues Rechteck zu zeichnen und dieses zu *Rechtforekeln*.“

Vier Tage später wurde das Beschreiben rekursiver Figuren wiederholt. Auch dort zeigte sich, dass das iterative Denken sehr stark ausgeprägt war.

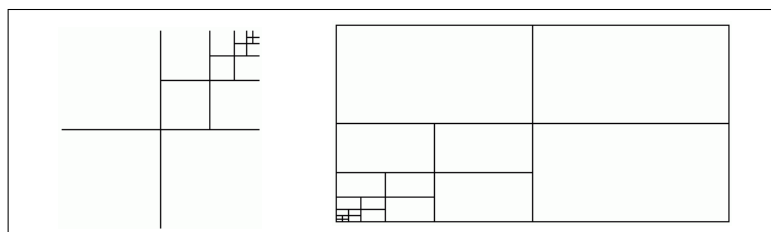


Abbildung 6.29: Rekursive Kreuz- und Rechtecksfiguren

26 Lösungen wurden von den Schülerinnen und Schülern für die beiden Figuren in Abbildung 6.29 erstellt, doch keine Formulierung erfolgte rekursiv. Zwar definierten neun Kinder einen Namen für das Vorgehen, verwendeten diesen dann aber nicht in ihrer Be-

schreibung. 14 gaben die Anzahl der ineinander geschachtelten Figuren genau an (vgl. Tabelle 6.26).

Formen der iterativen Beschreibung der Anzahl der Figuren	Anzahl
Jede einzelne Figur (z.B. „Male ein großes Kreuz und oben rechts vom Kreuz noch ein neues Kreuz und dann in das Kreuz oben rechts noch ein Kreuz und dann oben rechts noch ein Kreuz und dann wieder oben rechts noch ein Kreuz.“)	4
Konkrete Angabe von Wiederholungen (z.B. „Malen Sie ein Kreuz. In das Eck rechts oben malen Sie ein halb so großes Kreuz. Wiederholen Sie es 3 mal.“)	10
Offen (z.B. „Wiederhole das so oft wie es geht.“)	10
Sonstige	2
Größenangaben der geschachtelten Grundfigur	Anzahl
Keine	15
Halbierung (z.B. „ein um die Hälfte kleineres Kreuz“)	3
Sonstige Angaben (z.B. „malst du wieder ein Kreuz so groß, dass die ganze Ecke ausgefüllt ist“ oder „ein kleineres Kreuz“)	8

Tabelle 6.26: Beschreibung der rekursiven Kreuzfigur ($N = 26$)

Annähernd gleich verteilte sich die für die Rekursion wichtige offene Formulierung der Anzahl der Wiederholungen auf die Lösungen der korrespondierenden Gruppen.

Das Ziel, Schülerinnen und Schüler mit der Eleganz kompakter, rekursiver Beschreibungen zu faszinieren, wurde nach Ansicht der Forscherin allerdings erreicht, da die Kinder begeistert im Unterricht mitarbeiteten und selbstständig versuchten, rekursive Beschreibungen zu entwickeln. In weiteren Klassenstufen muss rekursives Denken noch geübt werden.

6.4.4 Informatische Beschreibungsmittel

Programmablaufplan

Problemlösungen können sich aus verschiedenen Einzelaktionen zusammensetzen, die in einer vorgegebenen Reihenfolge ausgeführt werden müssen. Für ihre Beschreibung eignen sich *Programmablaufpläne*, eine normierte Methode zur grafischen Darstellung von Ablaufstrukturen.

Einen relativ komplexen Programmablaufplan entwickelten die Schülerinnen und Schüler zum Vorgehen bei der Bildung des Komparativs regelmäßiger und auf „e“ endender Adjektive und *stark* und *gut* aus vorgegebenen Entscheidungen und Handlungen (vgl. Kapitel 4.1, S. 78 f.). Korrekt waren 14 Programmablaufpläne, elf enthielten Fehler und fünf Kinder lösten die Aufgabe überhaupt nicht.

Bei genauerer Betrachtung der Struktur der Programmablaufpläne fiel auf, dass Fehler gehäuft in Lösungen auftraten, in denen die Schülerinnen und Schüler die Entscheidungen und Handlungen völlig unstrukturiert angeordnet hatten (vgl. Anhang H, S. 249). In Lösungen, die inhaltlich die Allgemeinheit der Bildungsregeln der Komparativform berücksichtigten (vgl. Anhang H, S. 250), war die Quote der richtigen Lösungen höher. Es wird deutlich, dass sowohl formale als auch inhaltliche Strukturierungen zu besseren Ergebnissen führten. Vermutlich haben genau die im formallogischen Denken starken Kinder diese Fähigkeit und können die Aufgabe besser lösen (vgl. Abbildung 6.30). Auch die Kinder mit einem besseren kognitiven Gesamtleistungsniveau waren erfolgreicher.

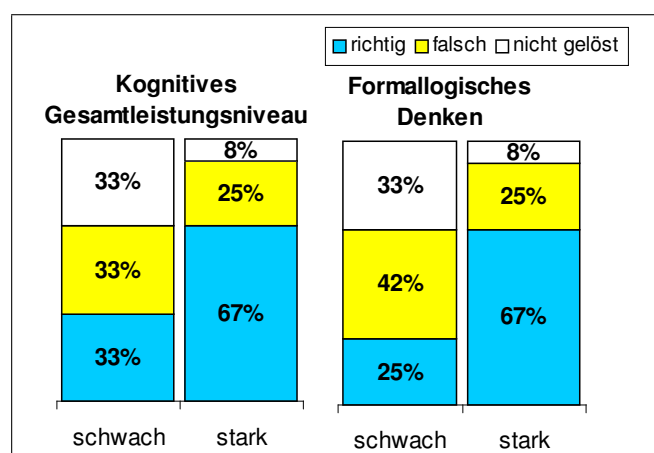


Abbildung 6.30: Erfolg bei der Erstellung des Programmablaufplans zur Steigerung von Adjektiven

Syntaxdiagramm

Syntaxdiagramme sind informatische Beschreibungsmittel zur Darstellung von kontextfreien Grammatiken (Claus und Schwill, 2003).

Im zweiten Test in Aufgabe 2 waren die Elemente des Syntaxdiagramms, d.h. die Satzglieder und der Punkt, schon vorgegeben (vgl. Anhang G, S. 244). Diese mussten für zwei Sätze in die richtige Reihenfolge gebracht werden. 23 Darstellungen waren korrekt. In den sieben fehlerhaften Syntaxdiagrammen vertauschten die Schülerinnen und Schüler die Reihenfolge der Dativ- und Akkusativobjekte.

Beim anschließenden Formulieren eines Beispielsatzes wurden die Satzglieder wieder in der Reihenfolge der Beispiele und nicht entsprechend des Syntaxdiagramms angeordnet.

Wahrscheinlich hatten die Kinder eher Schwierigkeiten mit der Bestimmung der Satzglieder als mit der Entwicklung der abstrakten Beschreibung in Form des Syntaxdiagramms. Die leistungstärkeren Schülerinnen und Schüler und diejenigen mit größeren Fähigkeiten beim zahlengebundenen Denken konnten die Aufgabe besser lösen (vgl. Abbildung 6.31).

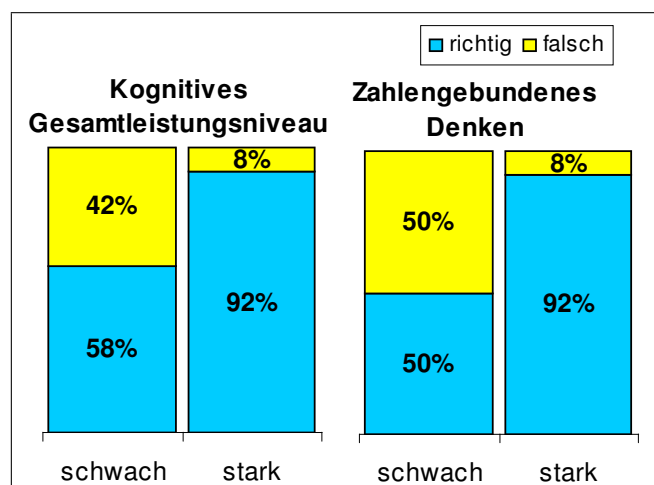


Abbildung 6.31: Erfolg bei der Erstellung eines Syntaxdiagramms für deutsche Sätze unter Vorgabe von Bausteinen

In Aufgabe 1 des Tests (vgl. Anhang G, S. 243) waren die Bausteine des Syntaxdiagramms nicht vorgegeben. Da nicht davon ausgegangen werden konnte, dass alle das Satzglied der adverbialen Bestimmung des Ortes kannten, wurde es in den beiden Beispielsätzen mit *Ort* gekennzeichnet. Die beiden Subjekte erhielten eine Marke *Blume*. Das Syntaxdiagramm der einfachsten Lösung hätte folgendermaßen ausgesehen:

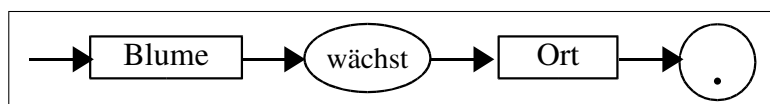


Abbildung 6.32: Syntaxdiagramm deutscher Sätze

21 Schülerinnen und Schüler nahmen die Hilfestellung überhaupt nicht wahr. Sie verwendeten in ihren Syntaxdiagrammen für die Blume 21 Mal *Subjekt* und den Ort 13 Mal *Objekt*, *Akkusativ*, *Dativ* oder richtigerweise viermal *adverbiale Bestimmung des Ortes*. Obwohl das Prädikat in beiden Sätzen *wächst* war, wurde es von 16 Kindern zum *Prädikat* abstrahiert. Dies wurde jedoch nicht als Fehler gewertet.

Nur zwölf Syntaxdiagramme waren korrekt, 16 falsch und zwei Schülerinnen und Schüler versuchten überhaupt nicht, die Aufgabe zu lösen. Das Ergebnis fiel schlechter aus, da der Schritt der Abstraktion selbst vollzogen werden musste. Teilweise wurde sehr ungenau gearbeitet, da die Struktur der deutschen Sätze für das inhaltliche Verständnis nicht

analysiert werden musste. Hier waren die Kinder, die im KFT insgesamt und im Bereich des sprachgebundenen Denkens besser waren, erfolgreicher (siehe Abbildung 6.33).

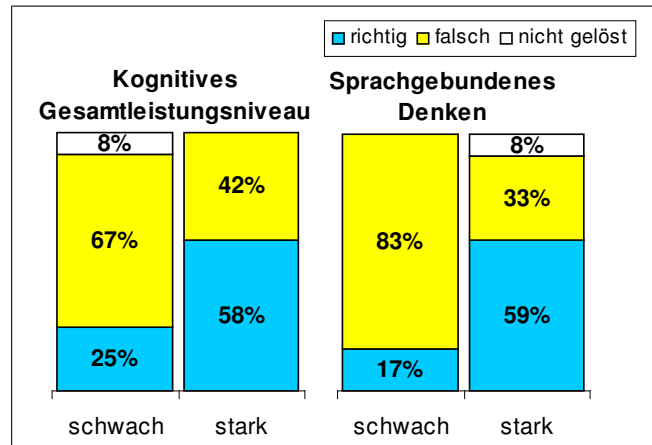


Abbildung 6.33: Erfolg bei der Erstellung eines Syntaxdiagramms für deutsche Sätze ohne Vorgabe von Bausteinen

Anders verhielt es sich mit englischen Sätzen, deren Inhalte nicht intuitiv erfasst werden konnten. Aus diesen leiteten 17 Schülerinnen und Schüler das Syntaxdiagramm richtig ab, obwohl ihnen die Satzstruktur zu diesem Zeitpunkt noch nicht bekannt war. Möglicherweise fällt eine allgemeine, abstrakte Darstellung von Sätzen in einer Fremdsprache leichter, da stärker in Strukturen und weniger in Inhalten gedacht wird. Fünf Kinder lösten die Aufgabe überhaupt nicht und acht Syntaxdiagramme waren fehlerhaft. Bei den englischen Sätzen schnitten die Kinder, die im KFT eine im zahlengebundenen und formallogischen Denken höhere Leistung erzielt hatten, besser ab (vgl. Abbildung 6.34).

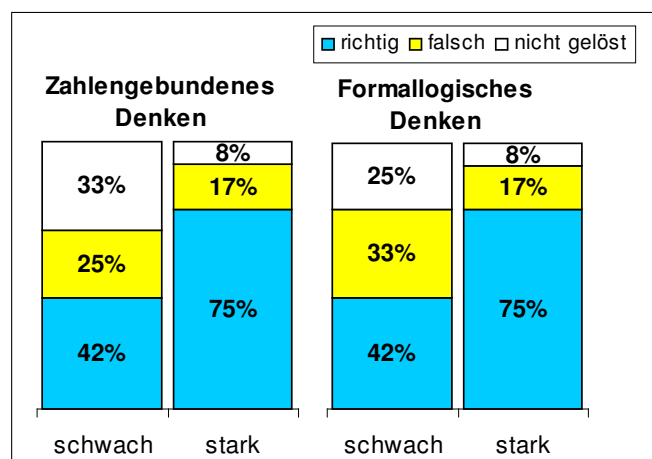


Abbildung 6.34: Erfolg bei der Erstellung eines Syntaxdiagramms für englische Sätze ohne Vorgabe von Bausteinen

UML-Klassendiagramm

UML-Klassendiagramme werden genutzt, um Klassen und ihre Beziehungen formal zu beschreiben. Die Beziehungen basieren auf den die Klassen beschreibenden Merkmalen.

In der Unterrichtseinheit zu den ebenen Figuren (vgl. Kapitel 4.1, S. 71 f.) mussten die Beziehungen zwischen den Klassen *Viereck*, *Parallelogramm*, *Rechteck*, *Raute* und *Quadrat* verbal in Form „jedes Quadrat ist auch ein Rechteck“ beschrieben werden. Natürlich existierten diese Beziehungen auch über mehrere Ebenen hinweg. In 12 der vorhandenen 26 Lösungen erkannten die Kinder dies und formulierten Sätze wie „Jede Raute ist ein Viereck und ein Parallelogramm“. In den 14 anderen Lösungen wurden nur die Beziehungen zur direkten Oberklasse beschrieben. Unabhängig vom Geschlecht und den im KFT ermittelten Fähigkeiten wurde die Aufgabe von allen Schülerinnen und Schülern annähernd gleich erfolgreich bearbeitet.

Im Test zu den informatischen Grundkonzepten wurden im Gegensatz zum UML-Klassendiagramm der ebenen Figuren die Attribute der übergeordneten Klassen nicht wiederholt (vgl. Anhang B, S. 201). Für vier Klassen der dort charakterisierten Lebewesen schrieben die Schülerinnen und Schüler die Eigenschaften auf. Nachdem im Vortest nur sieben Kinder die Aufgabe richtig lösten, waren es im Nachtest immerhin 13. Nur eine Schülerin, die die Aufgabe im Vortest korrekt bearbeitet hatte, löste sie im Nachtest nicht. Auch hier kann kein Zusammenhang mit dem Geschlecht und den Fähigkeiten aufgezeigt werden.

Hierarchien

Hierarchien sind Systeme, in denen Elemente einander über- und untergeordnet sind. Dabei darf jedes Element höchstens einem anderen untergeordnet sein.

Die Schülerinnen und Schüler mussten im Test zu den informatischen Grundkonzepten sechs Begriffe in eine vorgegebene Hierarchie mit acht Elementen einordnen (vgl. Anhang B, S. 202). Zwei der Elemente waren bereits eingetragen. Im Vortest lösten 13 und im Nachtest 12 Schülerinnen und Schüler die Aufgabe korrekt. Erstaunlich ist allerdings, dass sieben Kinder, die die Begriffe im Vortest korrekt in die Hierarchie eintrugen, im Nachtest die Aufgabe nicht richtig lösten. Ein möglicher Grund kann aus der Bearbeitungsweise der Aufgabe abgeleitet werden. Vor allem im Vortest waren die Schülerinnen und Schüler erfolgreicher, die die vorgegebenen Begriffe nach dem Einordnen in die Hierarchie systematisch markierten oder ausstrichen. Von den zwölf, die die Wörter kennzeichneten, lösten acht im Vortest die Aufgabe richtig. Im Nachtest war dieser Effekt nicht mehr so deutlich zu erkennen.

Zu Beginn des Schuljahrs unterschieden sich die Ergebnisse bei der Bearbeitung dieser Aufgabenstellung nur hinsichtlich der Variablen *formallogisches Denken* (vgl. Abbildung 6.35). Im Nachtest zeigte sich dagegen das bei der Gesamtauswertung des Tests

zu den informatischen Grundkonzepten analysierte Phänomen, dass am Ende des Schuljahrs die Jungen und die Kinder, die im sprachgebundenen Denken schwach und im zahlengebundenen und formallogischen Denken stark waren, bessere Resultate erzielten (vgl. Abbildung 6.36).

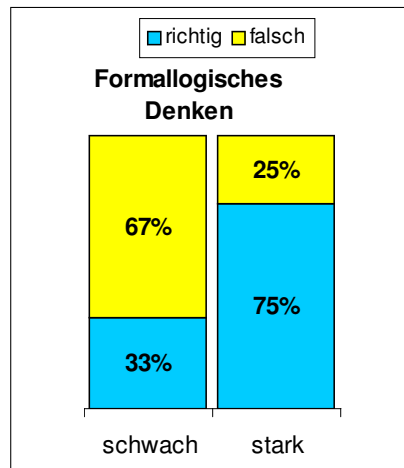


Abbildung 6.35: Erfolg bei der Bildung der Begriffshierarchie im Vortest

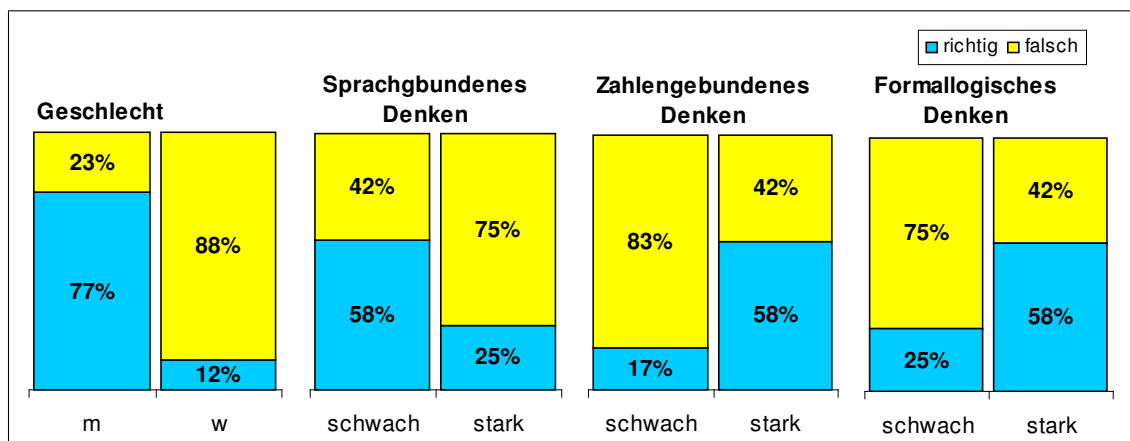


Abbildung 6.36: Erfolg bei der Bildung der Begriffshierarchie im Nachtest

Netzwerke

Im Gegensatz zu Hierarchien fehlt bei der Definition von *Netzwerken* die Bedingung, dass ein Knoten höchstens einem anderen unmittelbar untergeordnet sein darf. Vielmehr können die Knoten im Netzwerk beliebig miteinander verknüpft werden.

Zur Darstellung eines Stammbaums konstruierten die Schülerinnen und Schüler als Hausaufgabe ein Netzwerk (vgl. Kapitel 4.1, S. 91 f.). Der Stammbaum wurde durch 28 englische Sätze beschrieben. Da die Sätze nicht in der vorgegebenen Reihenfolge gelesen und

abgearbeitet werden konnten, stellte die Aufgabe eine sehr große Herausforderung für die Schülerinnen und Schüler dar. Aus diesem Grund versuchten nur 17 Kinder einen Lösungsansatz. Sieben von ihnen waren erfolgreich. Ihr Stammbaum enthielt alle Familienmitglieder in der korrekten Zuordnung der verwandtschaftlichen Beziehungen. Wie Abbildung 6.37 zeigt, waren die Schülerinnen und Schüler mit einem höheren kognitiven Gesamtleistungsniveau und besseren Ergebnissen im formallogischen Denken erfolgreicher.

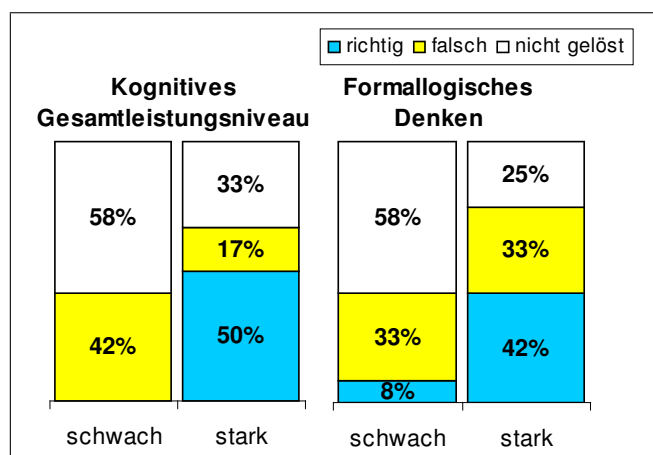


Abbildung 6.37: Erfolg bei der Ableitung eines Familienstammbaums

Ihre Lösungen enthielten sehr stark strukturierte Stammbäume, in denen die drei Ebenen der Großeltern, Eltern und Kinder deutlich erkennbar waren. Außerdem waren die Ehepaare häufig nebeneinander dargestellt und die Kinder mit Verbindungslinien den Eltern zugeordnet. Die Anordnung der Familienmitglieder in den fehlerhaften Lösungen war teilweise chaotisch (siehe Abbildung 6.38). Dazu kam noch ein nicht nachvollziehbares Netz von Verbindungslinien zur Kennzeichnung der verwandtschaftlichen Beziehungen.

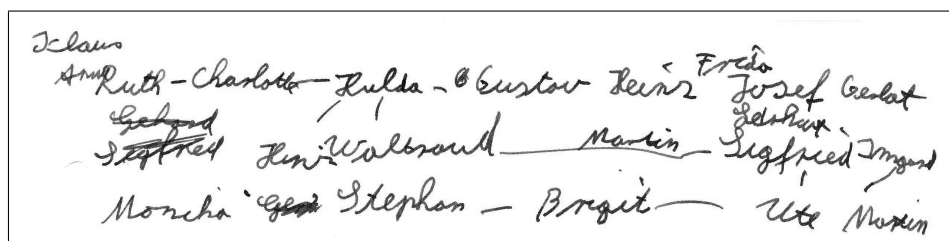


Abbildung 6.38: Chaotische Anordnung der Familienmitglieder in einem Stammbaum

Die räumliche Positionierung der Objekte zur Darstellung ihrer Zusammengehörigkeit und eine die Lesbarkeit unterstützende Nutzung von Verbindungslinien scheint also Voraussetzung für die Bildung von Netzwerken zu sein.

6.4.5 Fachnahe Inhalte

Grammatik

Nach Claus und Schwill (2003) ist eine *Grammatik* eine Menge von Regeln. Diese bestimmt, welche Sätze zu einer Sprache gehören und welche nicht.

Im Unterrichtsversuch definierten die Schülerinnen und Schüler in Deutsch und Englisch mehrmals für kleine Ausschnitte der Sprachen Grammatiken. In Englisch mussten sie, wie in Kapitel 4.1 (S. 84 f.) dargestellt, die Struktur von has/have-Sätzen und ihre Umformungen in den s-Genitiv abstrahieren. Obwohl das Syntaxdiagramm für has-Sätze als Beispiel vorgegeben war und die Satzteile in den Beispielsätzen in einem ersten Schritt farbig markiert worden waren, konnten nur zwölf Schülerinnen und Schüler die Grammatik für have-Sätze mit einem Syntaxdiagramm korrekt definieren. Sechs der acht falschen Lösungen verwendeten die flektierte Verbform „has got“ anstatt „have got“ und ein Schüler vergaß das Verb ganz. Der Transfer von den Sätzen im Singular auf den Plural war nicht möglich. Die Regeln wurden zu ungenau angegeben.

Vor allem die Schülerinnen und Schüler mit besseren allgemeinen und formallogischen Fähigkeiten zeichneten das Syntaxdiagramm korrekt (vgl. Abbildung 6.39).

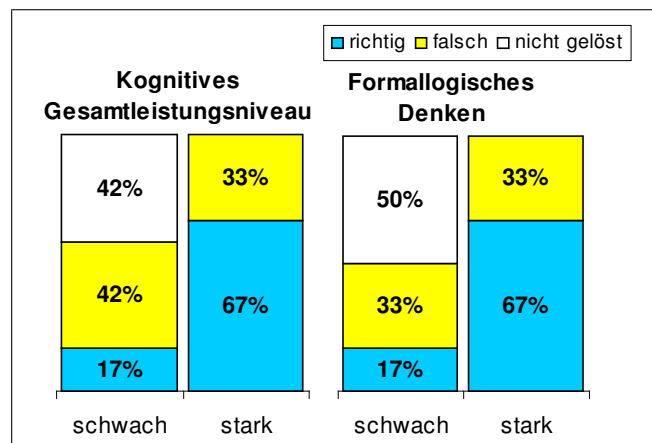


Abbildung 6.39: Erfolg bei der Darstellung der Grammatik für have-Sätze mit Hilfe von Syntaxdiagrammen

Das Syntaxdiagramm für die transformierten Sätze mit s-Genitiv konnten nur zwei Kinder nicht zeichnen. Obwohl die Satzglieder selbst bestimmt werden mussten und die Aufgabe dadurch schwieriger war, gelang 14 Schülerinnen und Schülern eine korrekte Darstellung. Die zuvor ausführliche Besprechung der Lösung der have-Sätze, der mehrfache Hinweis auf den Unterschied zwischen Terminal- und Nichtterminalsymbolen und die wiederholte Erklärung des Zusammenhangs zwischen der abstrakten Darstellung der Regeln einer Grammatik in Form von Syntaxdiagrammen und konkreten Sätzen trug sicherlich zu dem etwas besseren Ergebnis bei.

Zusätzlich zu den Gruppen, die schon bei der Erstellung des Syntaxdiagramms für have-Sätze erfolgreicher waren, erzielten auch diejenigen mit Stärken im zahlengebundenen Denken bessere Ergebnisse (vgl. Abbildung 6.40)

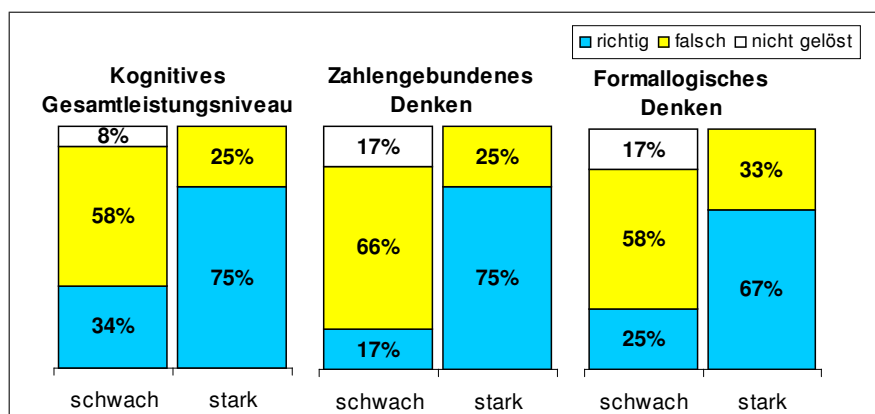


Abbildung 6.40: Erfolg bei der Darstellung der Grammatik für Genitiv-Sätze mit Hilfe von Syntaxdiagrammen

Syntax und Semantik

Die *Syntax* beschreibt den formalen Aufbau der Sätze einer Sprache, die *Semantik* hingegen die inhaltliche Bedeutung der Sprache.

Bei der Generierung der Sätze im Unterricht wurden nur die syntaktischen Strukturen beschrieben. Die Semantik konnte höchstens dahingehend beeinflusst werden, dass in den Generatoren der einzelnen Satzteile nur Elemente aufgenommen wurden, die auch zu semantisch korrekten Sätzen führten. Im Ritterprojekt bei der Generierung des Gedichts berücksichtigten dies alle Gruppen. Allerdings enthielt die Lösung eines Paares einen syntaktischen Fehler, da als Reime die Verben im Infinitiv und nicht passend zum Subjekt in der dritten Person Singular angegeben wurden. Dies war anscheinend bei der Generierung der Beispielveise nicht aufgefallen.

Funktionen

Eine *Funktion* drückt die Abhängigkeit einer Größe von einer oder mehreren anderen Größen aus.

Funktionen wurden propädeutisch für den Mathematikunterricht der Klassenstufe 8 in Maschinendarstellung mit optionalen Eingabemöglichkeiten und genau einer Ausgabe im Unterricht eingeführt. Die Kommunikation der Funktionen untereinander wurde abstrakt mit zusammengesetzten Maschinen dargestellt und in Logo durch die Verkettung der Funktionen mit Hilfe der **SENTENCE**-Funktion programmiert. In den Tests während des Schuljahrs mussten zusammengesetzte Maschinen konstruiert werden. Für Fragen wie „Is Ben the brother of Nick?“ (siehe Anhang G, S. 241) fanden im ersten Test

nur vier Kinder eine richtige Lösung. Etwas besser fiel das Ergebnis im zweiten Test aus. Immerhin neun Schülerinnen und Schüler zeichnen eine richtige zusammengesetzte Maschine für Sätze der Form „Die Mohnblume wächst auf dem Feld.“ (siehe Anhang G, S. 243). Anders als im Test zu den informatischen Grundkonzepten waren bei der Erstellung der zusammengesetzten Maschinen die Schülerinnen und Schüler, die im sprachgebundenen Denken im KFT bessere Ergebnisse erzielt hatten, erfolgreicher (vgl. Abbildung 6.41). Wie schon bei einigen anderen informatischen Grundkonzepten zuvor erreichten außerdem die sehr leistungsstarken Kinder und die Schülerinnen und Schüler mit guten Leistungen im formallogischen Denken bessere Ergebnisse.

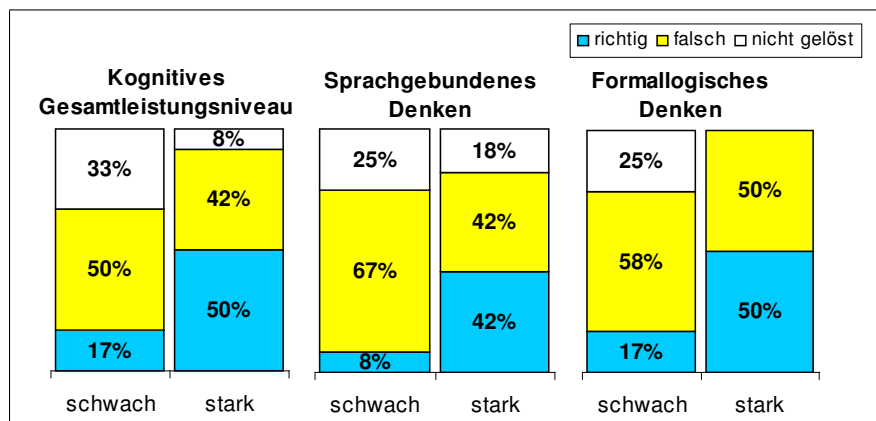


Abbildung 6.41: Erfolg bei der Erstellung der zusammengesetzten Maschinen zu englischen „Is“-Fragen

6.4.6 Zusammenfassung und Interpretation

Die Ergebnisse der Schülerinnen und Schüler, die aus der Analyse der Arbeitsblätter, Tests und Programme abgeleitet wurden, sind sehr heterogen. Die selbstständige Formulierung exakter Beschreibungen von Vorgehensweisen unter Berücksichtigung der Vollständigkeit und Konsistenz fällt teilweise noch schwer. Ein Grund dafür kann aus dem Verfahren bei der Abarbeitung vorgegebener Algorithmen abgeleitet werden. Anweisungen, die intuitive, aber notwendige Arbeitsschritte beinhalten, werden nur mental ausgeführt. Bei der eigenen Formulierung werden diese häufig vergessen, so dass die Beschreibungen unvollständig sind. Das Konzept der Automatisierung wird häufig bei der Erstellung eigener Bilder und Animationen angewendet. Vor allem mit dem **WIEDERHOLE**-Befehl wird sehr gerne bei der kreativen Erstellung von Bildern experimentiert. Am Ende des Schuljahrs können bereits 13 Schülerinnen und Schüler selbstständig einen Code entwickeln. Die Abbildungsvorschriften werden in sehr unterschiedlichen Formaten wie zum Beispiel mit Gleichheitszeichen, Bindestrichen, durch vertikale und horizontale Gegenüberstellungen, in Tabellen oder getrennt für Tag, Monat und Jahr verbal und in einer Zeichnung angegeben.

Die häufig erstellten Wortgeneratoren bei der Definition von Subsprachen enthalten immer wieder fehlerhafte Datenstrukturen. Dies ist sowohl in schriftlichen Tests als auch bei Arbeiten am Rechner der Fall, obwohl die Fehler dort direkt am Bildschirm sichtbar werden. Die Erweiterung der Modelle von Programmen gelingt allen Schülerinnen und Schülern. Interne Strukturen und Benutzerschnittstellen werden verändert und angepasst. Sicher können die Kinder die objektorientierten Konzepte des Aufrufs von Methoden, der Veränderung von Attributwerten und der Auswahl bzw. Benennung von Objekten sowohl an der Oberfläche des Imagine Logo Systems als auch auf programmiertechnischer Ebene anwenden.

Der Einsatz der Technik der Modularisierung und Hierarchisierung gelingt den Schülerinnen und Schülern in verschiedensten Kontexten. Die Abstraktion und die Formulierung dieser mit Hilfe der formalen informatischen Beschreibungsmittel der zusammengesetzten Maschinen oder der Syntaxdiagramme ist teilweise noch schwierig. Im Gegensatz dazu sind die Kinder beim Lesen der abstrakten Beschreibungen und der Ableitung von Beispielsätzen wesentlich erfolgreicher. Iterative Formulierungen in Logo bekommen schon sehr viele zustände. Teilweise muss noch darauf geachtet werden, dass keine redundanten Wiederholungen durchgeführt werden. Sehr stark unterscheiden sich die Schülerinnen und Schüler beim Zeichnen von Figuren, die rekursiv formuliert sind, und der eigenen Beschreibung rekursiver Figuren. Im ersten Fall ist die rekursive Struktur häufig erkennbar, auch wenn Details wie die Halbierung von Größen oder die genaue Platzierung vernachlässigt werden. Bei den eigenen verbalen Beschreibungen zeigt sich jedoch, dass das iterative Denken sehr stark ausgeprägt ist. Keine Ausführung erfolgt rekursiv und selbst die Anzahl der ineinander geschachtelten Figuren wird in den meisten Fällen konkret angegeben.

Eine wichtige Voraussetzung für korrekte formale Beschreibungen in Form von Programmablaufplänen oder Netzwerken scheint eine strukturierte Anordnung der Entscheidungen und Handlungen bzw. Knoten zu sein. Dies gelingt vor allem den allgemein Leistungsstärkeren und denjenigen mit Stärken im formallogischen Denken besser. Die Beschreibung von Satzstrukturen mit Hilfe von Syntaxdiagrammen fällt in Englisch besser als in Deutsch aus. Die Schülerinnen und Schüler denken in der Fremdsprache möglicherweise stärker in Strukturen und in der Muttersprache stehen die Inhalte im Vordergrund, da Strukturen intuitiv erfasst werden können. Die Beziehungen zwischen Klassen in UML-Klassendiagrammen über mehrere Ebenen hinweg können die Kinder nur zum Teil formulieren. Vor allem solange die informatischen Beschreibungsmittel noch nicht natürlich angewendet werden können, ist ein systematisches Vorgehen notwendig. Dies zeigt sich bei der Einordnung von Begriffen in Hierarchien. Die Schülerinnen und Schüler, die die vorgegebenen Begriffe nach dem Einordnen in die Hierarchie markieren oder durchstreichen, sind erfolgreicher.

Bei der Definition der Grammatik von Subsprachen und der Beschreibung der Syntax dieser mit Hilfe von Syntaxdiagrammen oder zusammengesetzten Maschinen fallen die

Ergebnisse sehr unterschiedlich aus. Wie bereits angedeutet, liegt dies eventuell an der mangelnden Abstraktionsfähigkeit. Ein Grund ist sicherlich auch die fehlende Konzentration der Schülerinnen und Schüler, da einfache Aufgabenstellungen aufgrund von Ungenauigkeiten teilweise schlechter als schwierige Aufgaben gelöst werden.

Insgesamt könnte der Eindruck entstehen, dass die informatischen Grundkonzepte in nur geringem Umfang erlernt werden, da oftmals weniger als die Hälfte der Schülerinnen und Schüler die analysierten Aufgaben vollständig und richtig lösen. Aber auch in anderen Unterrichtsfächern bewältigen Kinder nur einen Teil der gestellten Aufgaben, so dass dies der Normalität entspricht.

Die Ergebnisse der Auswertung der Unterrichtsmaterialien, getrennt nach den korrespondierenden Gruppen bezüglich der Merkmale *Geschlecht*, *kognitives Gesamtleistungsniveau*, *sprachgebundenes Denken*, *zahlungebundenes Denken* und *formallogisches Denken*, weichen nur zum Teil von denen aus den Tests zu den informatischen Grundkonzepten ab (vgl. Kapitel 6.3). Der signifikante Unterschied des Lernerfolgs, d.h. die Differenz der Ergebnisse des Nach- und Vortests, der Jungen und Mädchen spiegelt sich in der Bearbeitung der Aufgaben wider. In 5 der insgesamt 23 ausgewerteten Aufgabenstellungen haben bei den Jungen 30 Prozent mehr eine korrekte Lösung als bei den Mädchen. Umgekehrt ist dies nie der Fall (vgl. Tabelle 6.27). Der Unterschied ist allerdings nicht so groß wie im Test zu den informatischen Grundkonzepten. In Bezug auf das *kognitive Gesamtleistungsniveau* und das *zahlungebundene* und *formallogische Denken* ist der Effekt deutlicher. Überlappen die Interquartilbereiche der Ausprägungen „stark“ und „schwach“ in den Box-Plots der Differenzen der Ergebnisse des Nach- und Vortests (vgl. Abbildung 6.7) dieser Merkmale zu großen Teilen, so zeigt die qualitative Auswertung in 11, 8 bzw. 13 der Fälle deutlich bessere Ergebnisse in den Gruppen der Leistungsstärkeren (siehe Tabelle 6.27). Fähigkeiten im sprachgebundenen Denken spielen im Gegensatz zum Test zu den informatischen Grundkonzepten überhaupt keine Rolle.

Werden die Ergebnisse der quantitativen und qualitativen Auswertungen zusammengefasst, so wird deutlich, dass die Jungen und die insgesamt sehr leistungsstarken Schülerinnen und Schüler die Aufgaben zu den informatischen Grundkonzepten im Laufe des Schuljahrs erfolgreicher bearbeiten können. Auch das zahlungebundene und formallogische Denken scheint die Anwendung der informatischen Grundkonzepte zu unterstützen.

6.5 Veränderung der Einstellungen zum Computer

Die meisten Schülerinnen und Schüler verwendeten den Computer zu Hause bereits vor dem Unterrichtsversuch bei ganz verschiedenen Tätigkeiten. Wie aus der Befragung zu Beginn des Schuljahrs hervorgeht, war dies vor allem das Spielen. Sehr selten wurden mit dem Computer Texte geschrieben, Informationen im Internet gesucht, E-Mails verschickt oder Bilder gestaltet. Auch Lernprogramme wurden bereits eingesetzt. Aufgrund dieser

Erfolg in Abhängigkeit des	Häufigkeit
Geschlechts	
Jungen	5
Mädchen	0
kognitiven Gesamtleistungsniveaus	
stark	11
schwach	0
sprachgebundenen Denkens	
stark	3
schwach	3
zahlengebundenen Denkens	
stark	8
schwach	0
formallogischen Denkens	
stark	13
schwach	0

Tabelle 6.27: Zusammenfassung der Auswertung der Unterrichtsmaterialien

Erfahrungen und der Haltung der Eltern, Geschwister und Freunde hatten die Schülerinnen und Schüler bereits vor dem Unterrichtsversuch bestimmte Einstellungen zum Computer als technisches Gerät. Diese wurden durch die intensive Beschäftigung mit den informatischen Grundkonzepten beeinflusst.

Die Unterschiede zwischen der Vor- und Nachbefragung werden mit Hilfe von Box-Plots dargestellt, da die zentralen Tendenzen mit dem Median und dem ersten (Q1) und dritten Quartil (Q3), die die mittleren 50 Prozent der Werte begrenzen, und die Ausreißer dargestellt werden können. Zusätzlich werden teilweise die Meinungsänderungen der Schülerinnen und Schüler in Häufigkeitsverteilungen visualisiert.

Da die Fragebögen nicht immer vollständig ausgefüllt worden waren und am Schuljahresende sogar zwei Fragebögen nicht zurückgegeben worden waren, fehlten zu verschiedenen Items Antworten. In Anhang D sind die Antworthäufigkeiten bezüglich der einzelnen Items dargestellt. In der Auswertung wurden nur die Antworten der Items berücksichtigt, die sowohl in der Vor- als auch der Nachbefragung vorlagen.

Motivationale Aspekte bei der Verwendung des Computers

Die Fragen zur Motivation beziehen sich auf folgende drei Bereiche:

- Bedeutung von Wissen über den Computer
- Anwendungsmöglichkeiten des Computers
- Auswirkungen der Arbeit mit dem Computer auf die Denkfähigkeiten.

Die Klasse stimmte der allgemeinen Aussage, dass Wissen über Computer wichtig sei, sowohl zu Beginn als auch am Ende des Schuljahrs zu. Die Meinungsänderungen einzelner Kinder sind stärker ablehnend (vgl. linke Häufigkeitsverteilung in Abbildung 6.42). Die Mediane zu der persönlichen Frage, ob Wissen über Computer einzelnen etwas bringe, lagen in beiden Befragungen bei 4 (*stimmt eher*). Insgesamt änderten jedoch sehr viele Kinder ihre Meinung, zum Teil sogar im Bereich von zwei und mehr Marken.

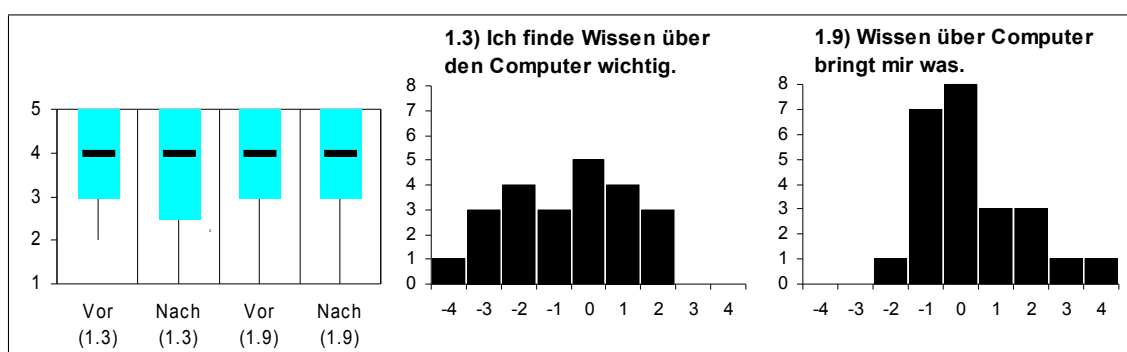


Abbildung 6.42: Veränderung der Einstellung zur Bedeutung von Wissen über den Computer (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Der Wunsch, die Funktionsweise des Computers kennen zu lernen (Item 1.8), ging im Laufe des Schuljahrs etwas zurück. Acht Schülerinnen und Schüler korrigierten ihre Einschätzung um zwei oder mehr Marken nach unten (siehe Häufigkeitsverteilung links in Abbildung 6.43). Auffälliger waren die Veränderungen bei der Frage, ob die Schülerinnen und Schüler mehr über den Computer wissen wollen (Item 1.6). Am Schuljahresbeginn stimmte die Klasse dieser Aussage fast vollständig zu. Das Interesse von 15 Kindern nahm bis an das Ende des Schuljahrs jedoch stark ab. Der Median sank von 5 (*stimmt voll*) auf 3.5 (zwischen *stimmt eher* und *teils teils*).

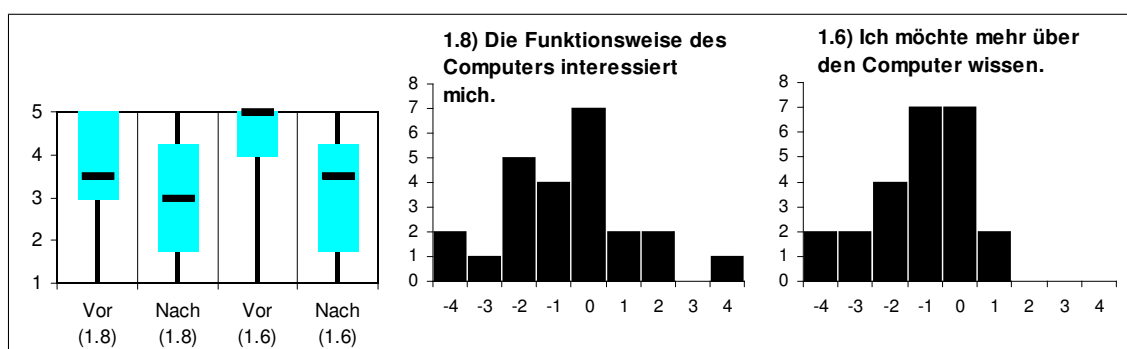


Abbildung 6.43: Veränderung der Einstellung zur Bedeutung von Wissen über den Computer (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Am wichtigsten war den Kindern auch nach dem Schuljahr noch das Spielen (vgl. Box-Plots in Abbildung 6.44). Der Median stieg von 3.5 (zwischen *teils teils* und *stimmt eher*)

auf 4 (*stimmt eher*) leicht an und ein sehr großer Teil der Kinder wählte in der Nachbefragung eine Marke mit einem höheren Zustimmungsgrad (vgl. linke Häufigkeitsverteilung in Abbildung 6.44). Insgesamt änderten bis auf sechs Kinder alle ihre Meinung.

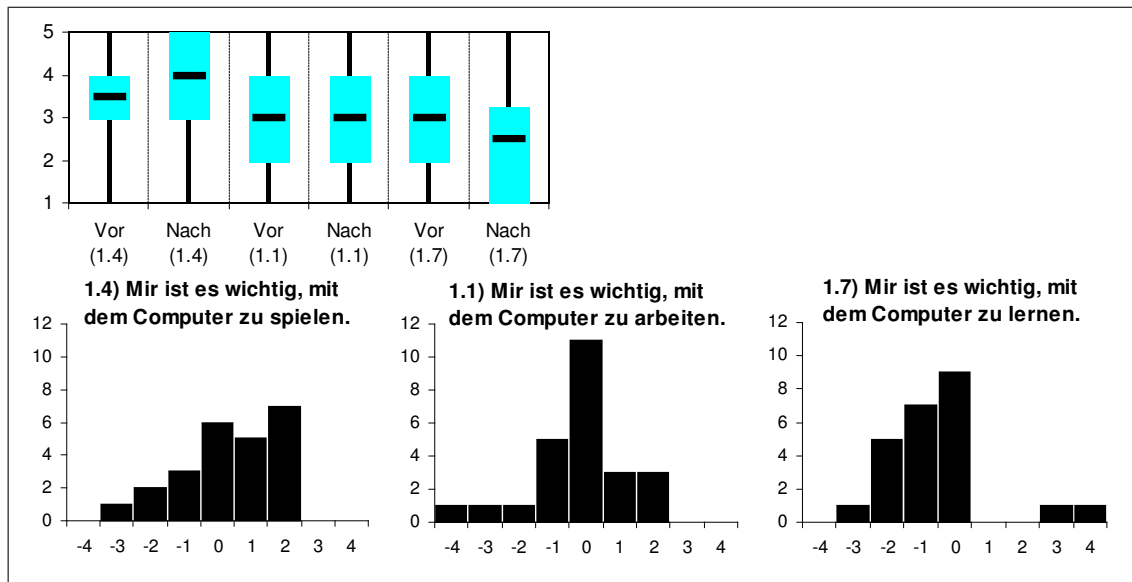


Abbildung 6.44: Veränderung der Einschätzung der Anwendungsmöglichkeiten des Computers (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Beim Item „Mir ist es wichtig, mit dem Computer zu arbeiten“ (Item 1.1) ist im Box-Plot keine Veränderung sichtbar. Bis auf zwei Fälle, deren Zustimmung um drei und vier Marken abnahm, zeigt auch die entsprechende Häufigkeitsverteilung in Abbildung 6.44 keine großen Schwankungen. Die Wichtigkeit, den Computer als Gerät zum Lernen einzusetzen, sank etwas. Der Interquartilbereich verschob sich nach unten. Nur für zwei Kinder wuchs die Bedeutung des Computers beim Lernen stark an.

Negative Auswirkungen auf die Denkfähigkeiten hat der Computer nach Meinung der Schülerinnen und Schüler nicht. Die Aussage, dass der Umgang mit dem Computer dumm mache (Item 1.2), lehnten fast alle ab (vgl. Box-Plots in Abbildung 6.45). Nur vier Kinder korrigierten ihre Meinung minimal. Dagegen divergierte die Einschätzung, ob der Umgang mit dem Computer das Denken fordere (Item 1.5), sehr stark. Sie sank, wie aus den Medianen, der Lage der Interquartilbereiche und der Häufigkeitsverteilung in Abbildung 6.45 hervorgeht, am Schuljahresende ab.

Zusammenfassung und Interpretation: Die Einschätzung der Bedeutung des Wissens über den Computer nahm im Laufe des Schuljahrs ab. Ein möglicher Grund dafür ist die inhaltliche Arbeit an den informatischen Grundkonzepten und in deren Kontext die Festigung der Anwenderfertigkeiten. Der Wissensbedarf ist nicht mehr so groß. Die Zunahme der Wichtigkeit des Spielens kann vielleicht auf den kognitiv fordernden Un-

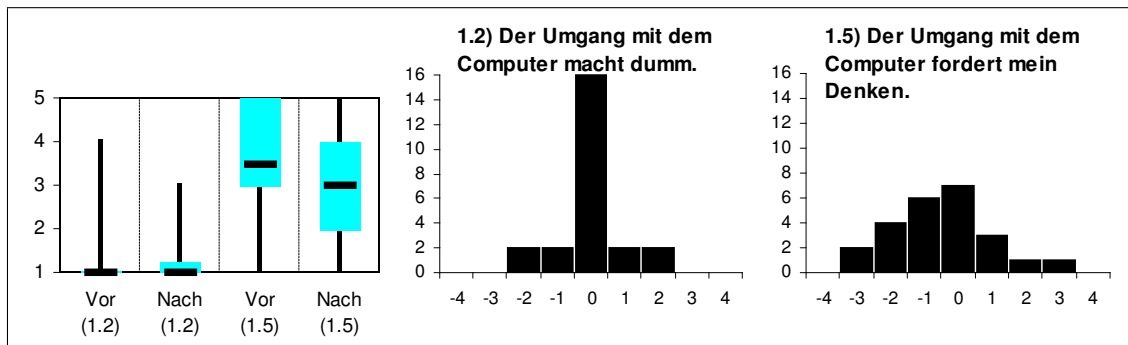


Abbildung 6.45: Veränderung der Einschätzung der Auswirkungen der Arbeit mit dem Computer auf die Denkfähigkeiten (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

terricht zurückgeführt werden. Die Schülerinnen und Schüler wünschten sich weniger anstrengende Tätigkeiten am Computer.

Gefühle bei der Benutzung des Computers

Die Schülerinnen und Schüler wurden auf der 5-stufigen Rating-Skala auch nach ihren Gefühlen bei der Benutzung des Computers gefragt. In Abbildung 6.46 sind die zentralen Tendenzen und ihre Verteilungsformen der Vor- und Nachbefragung in Box-Plots dargestellt. Die Mediane der Gefühle *Spaß*, *Interesse*, *Freude* und *Neugier* befanden sich im zustimmenden Bereich zwischen 5 (*stimmt voll*) und 3 (*teils teils*). *Anregung*, *Herausforderung* und *Entspannung* empfanden die Kinder eher weniger. Die Mediane und das erste und dritte Quartil lagen zwischen 1 (*stimmt überhaupt nicht*) und 3 (*teils teils*).

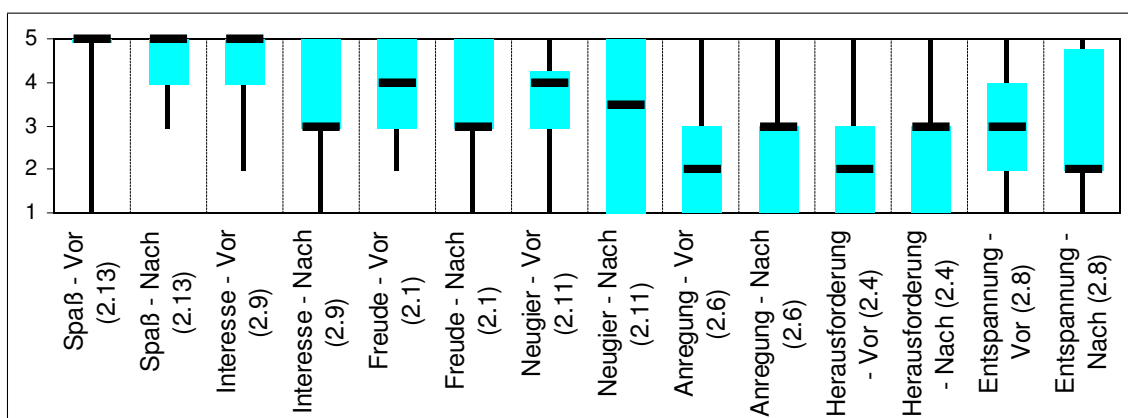


Abbildung 6.46: Box-Plots der positiven Gefühle bei der Benutzung des Computers (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Obwohl sich nur beim Item *Neugier* der Interquartilbereich stark veränderte, beurteilten die Kinder ihre Empfindungen in der Vor- und Nachbefragung häufig sehr unter-

schiedlich. Abgesehen vom Item *Spaß* wählten jeweils 52 bis 64 Prozent unterschiedliche Marken. Die Änderungen der Gefühle *Spaß*, *Freude*, *Anregung* und *Herausforderung* waren sehr ausgeglichen. Beim *Interesse*, der *Neugier* und der *Entspannung* zeigt Abbildung 6.47 hingegen, dass es ein Übergewicht hin zu einer stärkeren Ablehnung gab.

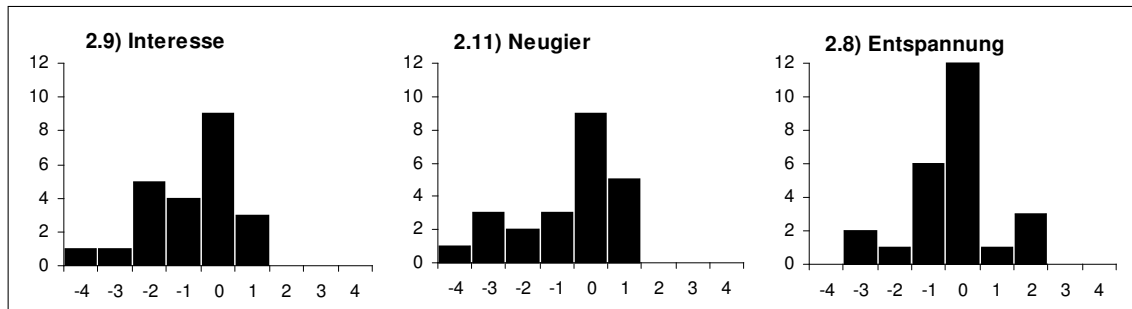


Abbildung 6.47: Veränderung des *Interesses*, der *Neugier* und der *Entspannung* bei der Benutzung des Computers

In den beiden Box-Plots des Items *Interesse* in Abbildung 6.46 wird dies durch die Abnahme des Medians von 5 (*stimmt voll*) auf 3 (*teils teils*) deutlich. Auch bei der *Neugier* sank der Median etwas. Auffälliger ist hier aber, dass sich das erste und dritte Quartil sehr stark veränderten. Dies zeigt, dass das Empfinden der Kinder in der Nachbefragung viel heterogener war. Eine weitere Streuung und ein Abfallen des Medians um eine Marke gab es auch beim Item *Entspannung*.

Negative Gefühle wie *Angst*, *Bedrohung*, *Unsicherheit*, *Beunruhigung* oder *Nervosität* äußerten die Schülerinnen und Schüler fast überhaupt nicht (vgl. Box-Plots in Abbildung 6.48).

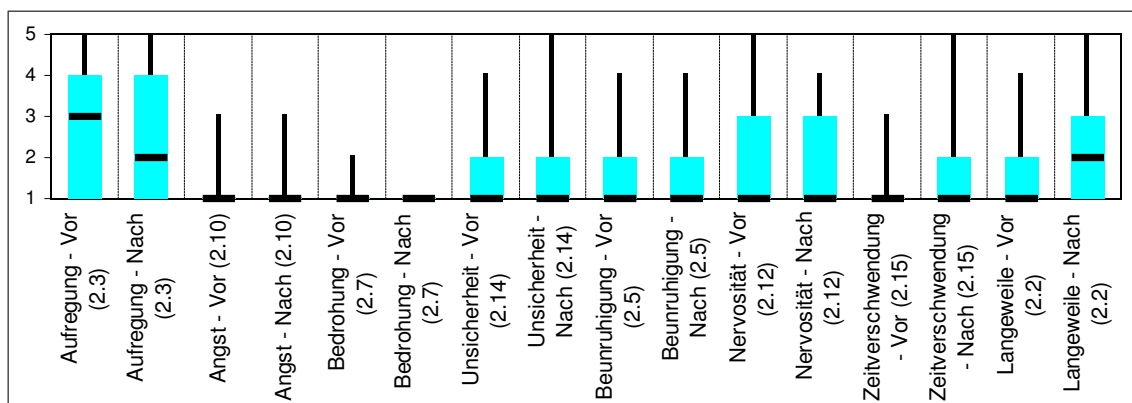


Abbildung 6.48: Box-Plots der negativen Gefühle bei der Benutzung des Computers (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Die Mediane lagen sowohl zu Beginn als auch am Ende des Schuljahrs bei 1 (*stimmt überhaupt nicht*). Im Fall der Items *Angst* und *Bedrohung* änderten auch nur 24 bzw.

4 Prozent der Kinder ihre Gefühlseinschätzungen. Ganz unterschiedlich war bei den Kindern die *Aufregung* ausgeprägt. Der Median sank insgesamt zwar von 3 (*teils teils*) auf 2 (*stimmt eher nicht*), das erste (Q1) und dritte Quartil (Q3) verblieben jedoch bei 1 (*stimmt überhaupt nicht*) und 4 (*stimmt eher*).

Allein bei den Items *Zeitverschwendung* und *Langeweile* veränderte sich der Interquartilsbereich. Wie in den beiden Häufigkeitsverteilungen der Meinungsänderungen in Abbildung 6.49 sichtbar ist, tendierten mehr Kinder dahin, die Empfindungen zu bestätigen.

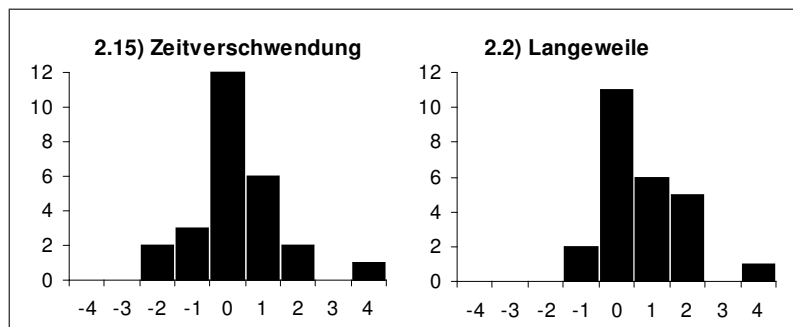


Abbildung 6.49: Veränderung des Gefühls der *Zeitverschwendung* und der *Langeweile* bei der Benutzung des Computers

Zusammenfassung und Interpretation: Die Gefühle und Empfindungen bei der Benutzung des Computers sind innerhalb der Klasse sehr stabil. Der Unterricht zu den informatischen Grundkonzepten beeinflusste sie also weder positiv noch negativ. Das Verhältnis der Kinder zum Computer ist sehr positiv. Sie haben Spaß und Freude und sind interessiert und neugierig. Die Benutzung des Computers stellt für sie keine Herausforderung dar und ängstigende Gefühle im Umgang mit dem Gerät kennen sie überhaupt nicht.

Selbsteinschätzung der Kompetenzen im Umgang mit dem Computer

Ihre Fähigkeiten im Umgang mit dem Computer schätzte die Mehrheit der Schülerinnen und Schüler schon zu Beginn des Schuljahrs sehr hoch ein (vgl. Box-Plots in Abbildung 6.50). Der Median des Items 3.4 („Der Umgang mit dem Computer fällt mir nicht schwer“) lag bei 4 (*stimmt eher*) und der des Items 3.1 („Ich traue mir im Umgang mit Computern sehr viel zu“) bei 3 (*teils teils*). Im letzteren Fall lagen die Antworten von 75 Prozent der Schülerinnen und Schüler allerdings im Bereich zwischen 3 (*teils teils*) und 5 (*stimmt voll*). Dies änderte sich auch in der Nachbefragung nicht. Aus der Häufigkeitsverteilung der Differenzen zwischen der Vor- und Nachbefragung in Abbildung 6.50 geht hervor, dass es nur leichte Meinungsschwankungen gab, die in ablehnender und zustimmender Richtung fast gleichgewichtet waren. Auffällig ist, dass die Selbsteinschätzungen der Schülerinnen und Schüler sehr stabil blieben. Es gab bei einzelnen Kindern maximal Abweichungen von zwei Marken.

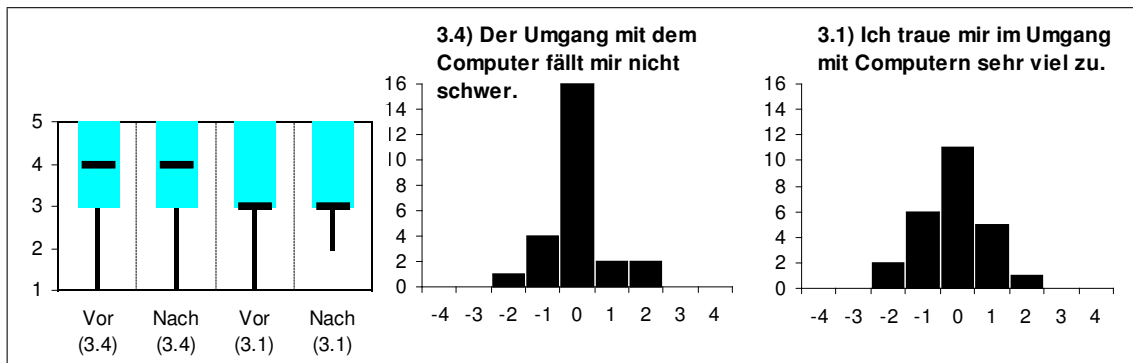


Abbildung 6.50: Veränderung der positiven Selbsteinschätzung im Umgang mit dem Computer (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Die sehr pessimistischen Selbsteinschätzungen in den Items 3.2, 3.3 und 3.5 in Bezug auf die eigenen Fähigkeiten, mit dem Computer umzugehen, lehnten die Schülerinnen und Schüler zum großen Teil sowohl in der Vor- als auch in der Nachbefragung vollständig ab (vgl. Box-Plots in Abbildung 6.51). Nur in der Nachbefragung stimmte ein Kind dem Item „Der Computer liegt mir nicht“ (Item 3.2) voll zu. Ansonsten lagen die negativsten Selbsteinschätzungen bei 3 (*teils teils*) oder 4 (*stimmt eher*).

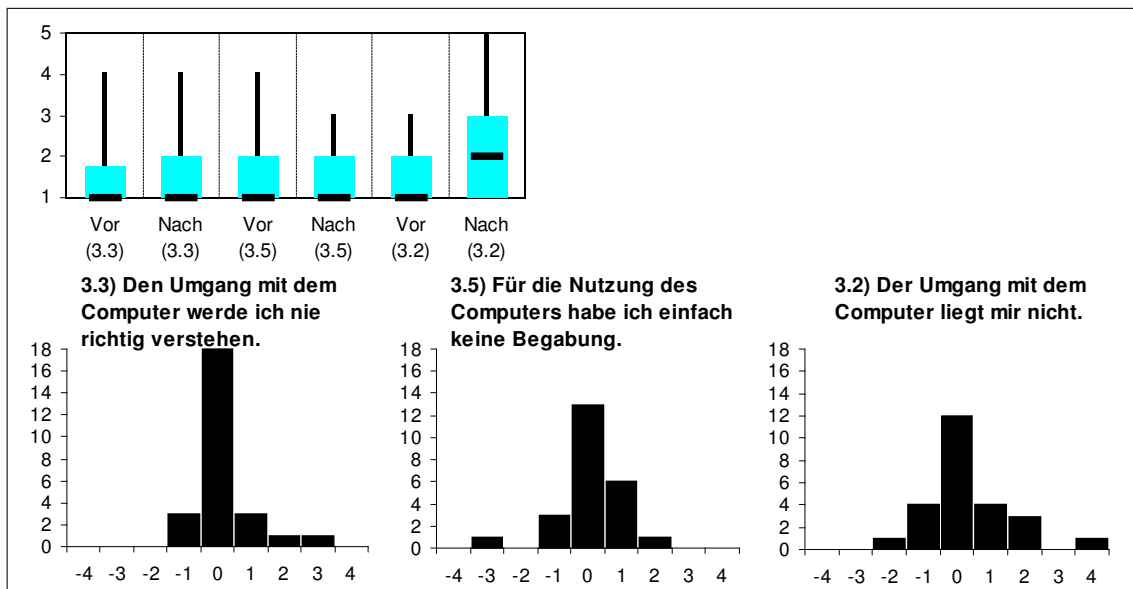


Abbildung 6.51: Veränderung der negativen Selbsteinschätzung im Umgang mit dem Computer (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Zusammenfassung und Interpretation: Die Schülerinnen und Schüler sind bezüglich ihrer Leistungen im Umgang mit dem Computer sehr selbstbewusst. Durch die sich von zu Hause unterscheidende Arbeitsweise im Unterricht zu den informatischen Grund-

konzepten, die sehr anspruchsvoll ist, verschlechtert sich wahrscheinlich bei manchen Kindern die Einschätzung der eigenen Fähigkeiten am Schuljahresende.

Beurteilung der Eigenschaften des Computers

Am Ende des Fragebogens charakterisierten die Schülerinnen und Schüler die Eigenschaften des Computers. Realistisch schätzten sie ein, dass der Computer nur das arbeiten könne, was in dem benutzten Programm stehe (Item 4.6) und dass ihm genau gesagt werden müsse, was er tun solle (Item 4.3). Im ersten Fall veränderten sich die Meinungen über das Schuljahr hinweg in beide Richtungen (siehe Häufigkeitsverteilung der Meinungsänderungen zu Item 4.6 in Abbildung 6.52). Auf Grund der eigenen Erfahrungen, mit Logo sehr exakt arbeiten zu müssen, korrigierten acht Kinder ihre Einschätzung bezüglich der Notwendigkeit zum genauen Arbeiten mit dem Computer (Item 4.3) nach oben.

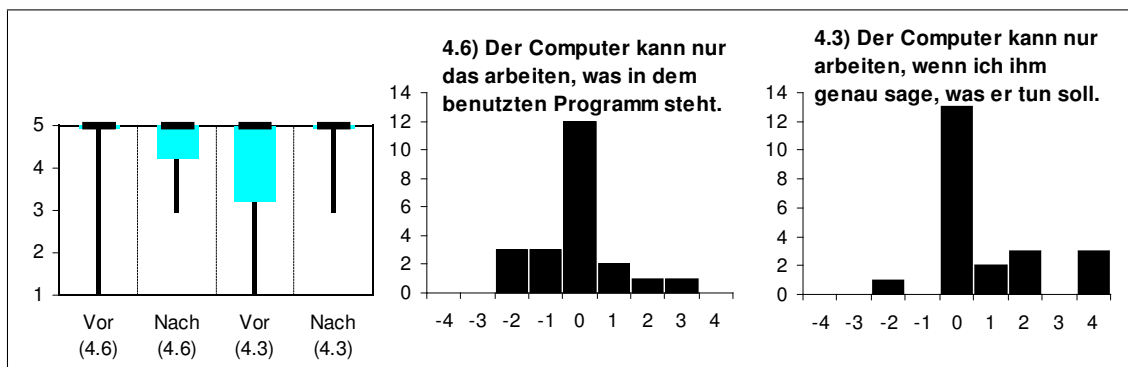


Abbildung 6.52: Veränderung der Einschätzung der Eigenschaften des Computers
(1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Als „böartig“ empfanden die Kinder den Computer nicht. Diese Einschätzung änderte sich auch nicht am Ende des Schuljahrs (vgl. Abbildung 6.53).

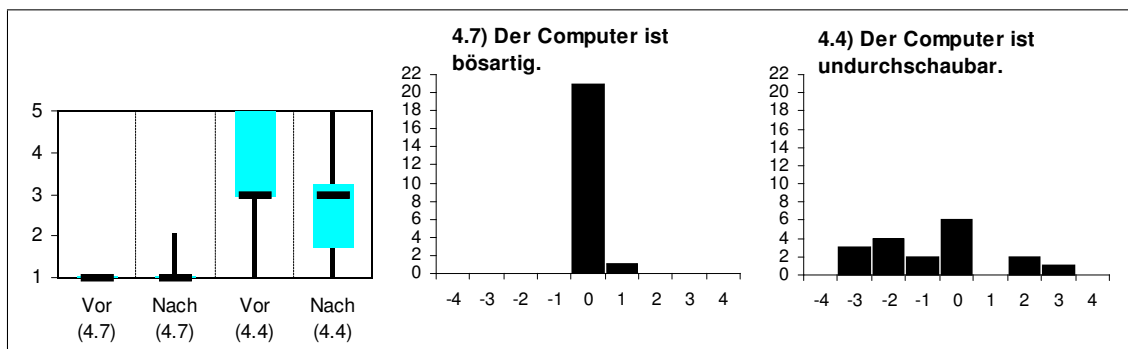


Abbildung 6.53: Veränderung der Einschätzung der Eigenschaften des Computers
(1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Die Undurchschaubarkeit wurde sehr heterogen beurteilt. Der Median lag in beiden Befragungen bei 3 (*teils teils*), der Interquartilbereich sank jedoch stark ab. Auch in der Häufigkeitsverteilung der Meinungsänderungen (siehe rechtes Diagramm in Abbildung 6.53) ist diese Tendenz deutlich zu erkennen. Wesentlich mehr Schülerinnen und Schüler berichtigten ihre Zustimmung nach unten als noch oben.

Sicher waren sich die Schülerinnen und Schüler darin, dass der Computer sie bei Problemen nicht verstehe (Item 4.2) und dass er ihnen selbstständig keine Hilfe anbiete (Item 4.5). Die Mediane blieben bei 1 (*stimmt überhaupt nicht*). Das dritte Quartil (Q3) sank am Schuljahresende ebenfalls auf 1. Die Häufigkeitsverteilungen in Abbildung 6.54 zeigen, dass die Meinungsänderungen hauptsächlich in Richtung Ablehnung tendierten.

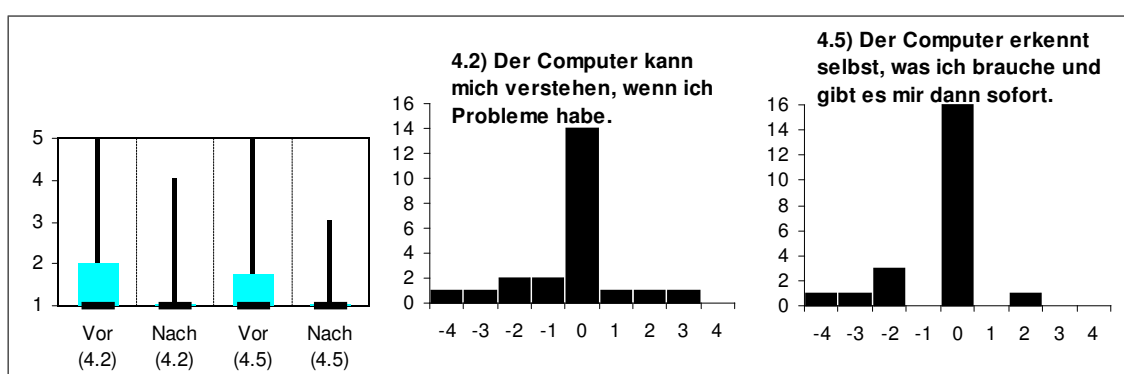


Abbildung 6.54: Veränderung der Fehlvorstellungen über Eigenschaften des Computers (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Hartnäckig hielt sich bei wenigen Schülerinnen und Schülern die Tendenz in der Einschätzung, dass der Computer ihnen teilweise das Lernen abnehme (Item 4.10). Sieben Schülerinnen und Schüler korrigierten diese sogar in Richtung Zustimmung (vgl. linke Häufigkeitsverteilung der Meinungsänderungen in Abbildung 6.55).

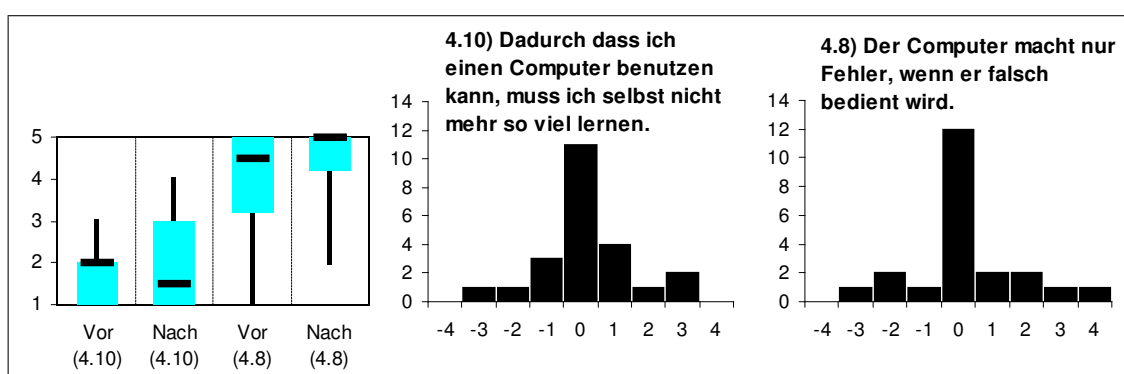


Abbildung 6.55: Veränderung der Fehlvorstellungen über Eigenschaften des Computers (1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Obwohl die Kinder im Unterricht selbst Programme erstellten und wussten, dass sie fehlerhaft sein können, stimmten sie sowohl zu Beginn als auch am Ende des Schuljahrs voll darin überein, dass der Computer nur Fehler macht, wenn er falsch bedient wird (Item 4.8). In der Nachbefragung gab es keine Schülerin und keinen Schüler, der diese falsche Aussage völlig ablehnte (siehe Box-Plot in Abbildung 6.55).

In ihrer Einschätzung, dass der Computer alles kann, waren die Kinder eher vorsichtig. Der Interquartilbereich lag in der Vor- und Nachbefragung zwischen 1 (*stimmt überhaupt nicht*) und 3 (*teils teils*) (vgl. Box-Plot in Abbildung 6.56) und der Median sank von 2 (*stimmt eher nicht*) auf 1 (*stimmt überhaupt nicht*). Die in der Häufigkeitsverteilung dargestellten Meinungsschwankungen sind sehr groß. Lediglich acht Schülerinnen und Schüler blieben am Schuljahresende bei ihrer Aussage. Im Gegensatz dazu stimmte die Mehrheit der Kinder darin überein, dass der Computer schnell rechnen kann.

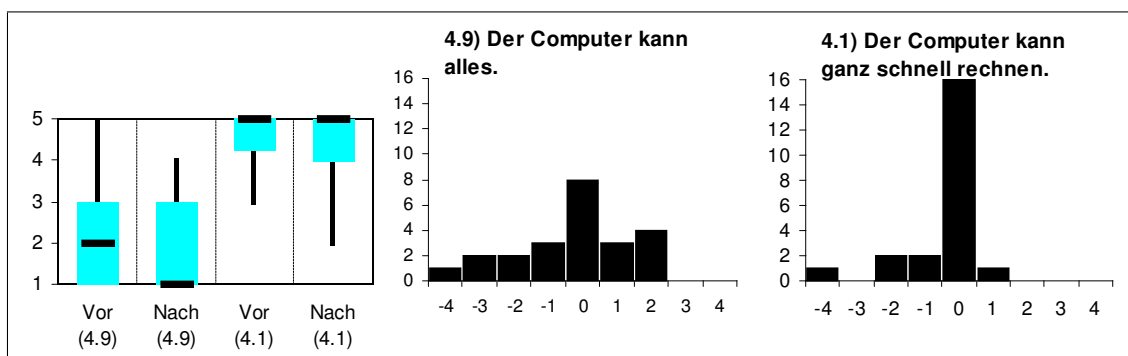


Abbildung 6.56: Veränderung der Einschätzung der Fähigkeiten des Computers
(1 = *stimmt überhaupt nicht*, 5 = *stimmt voll*)

Zusammenfassung und Interpretation: Den Schülerinnen und Schülern ist klar, dass der Computer nicht selbstständig arbeitet, sondern nur Anweisungen eines Programms oder einer Benutzerinteraktion folgen kann. Außerdem erhoffen sie sich von ihm bei Problemen keine Hilfestellungen. Das Gefühl der Kinder, dass der Computer undurchschaubar ist, hat bei einigen am Ende des Schuljahrs abgenommen. Dies kann möglicherweise auf die intensive Beschäftigung mit dem Computer zurückgeführt werden, die sich systematisch an einigen informatischen Grundkonzepten orientierte. Beharrlich bleibt bei einigen Kindern die Vorstellung erhalten, dass sie durch die Benutzung des Computers selbst nicht mehr so viel lernen müssten. Bei wenigen nimmt sie sogar noch zu. Völlig unerklärlich ist die Annahme fast aller Schülerinnen und Schüler, dass der Computer nur Fehler mache, wenn er falsch bedient würde. Mehrfach hatten die Kinder im Unterricht selbst erfahren, dass ihre Logo-Programme „Fehler“ machten, weil sie von ihnen selbst nicht korrekt erstellt worden waren. Möglicherweise können sie hier auch nicht zwischen der Bedienung von Computern und dem Schreiben von Programmen unterscheiden. Realistisch hingegen ist ihre Einschätzung, dass der Computer nicht alles kann.

7 Fazit

In der vorliegenden Arbeit wurde herausgearbeitet, dass die in den *Bildungsstandards zur Informationstechnischen Grundbildung (ITG)* im Bildungsplan 2004 für Realschule (Kultusministerium Baden-Württemberg, 2004b) beschriebenen informationstechnischen Kompetenzen und Inhalte den geforderten Ansprüchen einer Allgemeinbildung nicht genügen, sondern allgemeine informatische Grundkonzepte vermittelt werden müssen. Im Rahmen des Forschungsprojekts wurde ein Unterrichtskonzept für die Klassenstufe 5 entwickelt, in dem die informatischen Grundkonzepte, eingebettet in den Kontext der Fächer Mathematik, Englisch, Deutsch und Musik, in vier Stunden der Woche altersgemäß vermittelt werden. Der Computer wird dort unter ausschließlicher Verwendung von Logo-Software als zwangloses Denk-, Hilfs- und Arbeitsmittel zum Erlernen der Fachinhalte und informatischen Grundkonzepte eingesetzt. In den meist vorgegebenen mathematischen, linguistischen und musikalischen Logo-Mikrowelten haben die Schülerinnen und Schüler die Möglichkeit, Lösungen zu Problemstellungen aus dem Fachunterricht zu gestalten.

7.1 Zusammenfassung

Mit der vorgelegten Untersuchung konnte gezeigt werden, dass die Vermittlung informatischer Grundkonzepte bereits in Klassenstufe 5 im Fachunterricht möglich ist. Die Schülerinnen und Schüler sind trotz der teilweise abstrakten informatischen Inhalte, des gleichzeitigen Lernens von Fachinhalten und von informatischen Grundkonzepten, der Umsetzung von Problemlösungen in die Programmiersprache Logo und des häufig fachintegrativen und fächerübergreifenden Lernens und Arbeitens nicht überfordert. Der kognitiv sehr anspruchsvolle Unterricht zu den informatischen Grundkonzepten ist für Schülerinnen und Schüler nicht schwieriger als der Fachunterricht. Die Fortführung des Unterrichts ist erwünscht.

Als geeignet erweist sich die Aufteilung des Schuljahrs in drei Phasen, die sich bezüglich der Methoden, der Wissensvermittlung und -anwendung und der Integration der informatischen Grundkonzepte in die Fachinhalte unterscheiden. Zu Beginn ist es sinnvoll, die informatischen Grundkonzepte und Logo-Sprachkonstrukte in kürzeren Unterrichtseinheiten eingebettet in die Fachinhalte zu vermitteln. Die abstrakten Inhalte können schrittweise eingeführt und die Vernetzung der informatischen Grundkonzepte und Fachinhalte

auf ein zu bewältigendes Maß eingeschränkt werden. Im anschließenden fächerübergreifenden Projektunterricht wird bei der Erstellung einer multimedialen Rittergeschichte das zuvor Erlernte angewendet und vertieft. Die Schülerinnen und Schüler arbeiten sehr motiviert an Themen aus verschiedenen Fächern, die implizit mit Kenntnissen der informatischen Grundkonzepte verknüpft sind. Aufgrund der Möglichkeit, sich mit seinen speziellen Fähigkeiten einbringen zu können, entstehen imponierende fächerübergreifende Arbeiten. Mit Problemstellungen, die nicht nur fachlich begründet sind, sondern sich an den Interessen der Schülerinnen und Schüler orientieren, kann das Schuljahr inhaltlich abgerundet werden. Bei der Erweiterung von Spielen und eines Malprogramms werden eigene aktive Gestaltungsmöglichkeiten bei der Arbeit mit einer Programmierumgebung deutlich und informatische Grundkonzepte des Programmierens im Großen erlernt und vertieft.

Die Motivation der Schülerinnen und Schüler ist nach dem Abklingen des Neuigkeitseffekts durch den Einsatz des Computers normal. Sie hängt wie auch in anderen Fächern sehr stark von den individuellen Interessen, Einstellungen und Fähigkeiten der einzelnen Kinder ab. Motivationshöhen sind die Unterrichtseinheiten, in denen sehr viel praktisch gearbeitet wird. In zu langen Theoriephasen im Klassenzimmer darf die Ausdauer der Schülerinnen und Schüler nicht überstrapaziert werden, um zu vermeiden, dass sich eine Polarisierung des Unterrichts in angenehme praktische Stunden im Computerraum und unangenehme theoretische im Klassenzimmer ergibt. Der Theorieunterricht muss somit auch im Computerraum stattfinden, so dass in kürzeren Phasen ein Wechsel mit praktischen Arbeiten am Rechner möglich ist. Die Form der Zusammenarbeit an den Rechnern gestaltet sich sehr unterschiedlich. Schwierigkeiten gibt es vor allem bei Jungen, da oftmals nur derjenige Partner geistig aktiv ist, der die Tastatur und Maus bedient. Andererseits arbeiten auch Paare gemeinsam, die sich aufgrund ihrer eher gestalterischen und technischen Fähigkeiten optimal in der Zusammenarbeit ergänzen. Kommunikation zwischen Schülerinnen und Schülern, die nicht an einem Computer zusammenarbeiten, findet kaum statt. Zur Stärkung der Kooperation innerhalb des Klassenverbandes müssen Aufgabenformen entwickelt werden, deren Lösung die Kommunikation erfordert. Vorbild sollte das Vorgehen bei der arbeitsteiligen Entwicklung von Softwaresystemen sein. Die Kinder hätten dann zum Beispiel die Aufgabe, Arbeitspakete zu vereinbaren, sich über Schnittstellen zu einigen und das Endprodukt aus den Teillösungen zusammenzusetzen.

Die Vermittlung der informatischen Grundkonzepte hat keine negativen Rückwirkungen auf den Fachunterricht. Alle Fachinhalte der Klassenstufe 5 in Mathematik, Englisch, Deutsch und Musik können vermittelt werden, obwohl wegen der zusätzlichen informatischen Grundkonzepte insgesamt weniger Unterrichtszeit zur Verfügung steht. In Englisch unterstützt die Beschäftigung mit den informatischen Grundkonzepten sogar das Erlernen der grammatikalischen Strukturen der Sprache. Außerdem werden durch die Thematisierung der informatischen Grundkonzepte in den verschiedenen Anwendungskontexten der beteiligten Fächer die Gemeinsamkeiten und Zusammenhänge der Fachinhalte deutlicher aufgezeigt.

Schülerinnen und Schüler können auch ohne formalen Unterricht vereinzelt informatische Strategien anwenden. Nach dem Unterricht zu den informatischen Grundkonzepten sind die Leistungen jedoch signifikant besser. In Klassenstufe 5 können also abstrakte informatische Inhalte erlernt werden. Die Leistungssteigerungen sind sehr unterschiedlich. Bei den Jungen und Mädchen unterscheiden sie sich sogar signifikant. Dies wird auch in den ausgewerteten Unterrichtsmaterialien deutlich. Zwischen den Ausprägungen „stark“ und „schwach“ der Merkmale *kognitives Gesamtleistungsniveau*, *sprachgebundenes Denken*, *zahlengebundenes Denken* und *formallogisches Denken* ergeben sich keine signifikanten Unterschiede bezüglich der Differenz der Ergebnisse des Nach- und Vortests. Die Bearbeitung der Aufgaben im Unterricht gelingt Kindern mit einem hohen *kognitiven Gesamtleistungsniveau* und stärkeren Ergebnissen im *zahlengebundenen* und *formallogischen Denken* besser. Um vor allem den schwächeren Kindern einen leichteren Zugang zu den theoretischen Konzepten anzubieten, müssten diese für sie „erlebbar“ werden. Eine Möglichkeit ist die Entwicklung weiterer Mikrowelten für die Themenbereiche, die ansonsten theoretisch bearbeitet werden. Dies hätte auch zur Konsequenz, dass die Kinder mehr am Computer arbeiten und die langen Theoriephasen verkürzt würden.

Ein äußerst wichtiges Ergebnis ist die Tatsache, dass im Unterricht allgemeine Kompetenzen wie das Vorausdenken, das Strukturieren, das Anwenden von Schemata, das genaue Arbeiten und Strategien zur Bewältigung von Frustrationen erworben werden. Diese Schlüsselqualifikationen sind nicht nur für eine informationstechnische Grundbildung, sondern in allen Bereichen des Lebens von enormer Bedeutung und werden wie bereits beschrieben zusätzlich zu den Fachinhalten erlernt. Natürlich wird auch die Bedienkompetenz von Informatiksystemen, die nicht explizit geschult wird, durch die Arbeit am Computer, aber auch durch die Vermittlung von informatischen Vorstellungen verbessert.

Die ausschließliche Verwendung der beiden Logo-Systeme fördert die Konzentration auf die informatischen Grundkonzepte, da eine explizite Einarbeitung nicht erforderlich ist. Sie erfolgt anhand von Aufgabenstellungen und nimmt nur einen ganz geringen Teil der Unterrichtszeit in Anspruch. Da die informatischen Grundkonzepte immer in einem Fachzusammenhang stehen, bleiben sie für die Kinder keine abstrakten Konstrukte. Ihre Umsetzung in Logo-Mikrowelten findet nicht in Form eines Programmierkurses statt, sondern die Schülerinnen und Schüler gestalten in mathematischen, linguistischen und musikalischen „Umwelten“ (Mikrowelten) Lösungen zu Problemstellungen aus dem Fachunterricht. Beim Programmieren trainieren sie das aktive Auseinandersetzen mit Problemen, Planen von Lösungen, Antizipieren von Situationen, Gliedern von komplexen Problemen, Abstrahieren, Formalisieren, präzises Denken und sprachliches Formulieren von Gedanken, exaktes Arbeiten, Denken in Querbezügen und das kritische Überprüfen der eigenen Arbeitsergebnisse (Mittermeir 2003, Sperber 2002). Dies sind genau die von den Lehrerinnen zum Teil beobachteten allgemeinen Schlüsselkompetenzen. Notwendige Voraussetzung für ihren Erwerb ist der Einsatz einer Programmiersprache wie Logo,

die einen geringen Sprachumfang besitzt, erweiterbar ist, mit der von der ersten Stunde an Problemlösungen erstellt werden können und die informatische Denkkonzepte veranschaulicht.

Die Einstellungen der Kinder zum Computer verändern sich durch den Unterricht zu den informatischen Grundkonzepten kaum. Die Bedeutung des Rechners im Alltag ist ihnen bewusst. Ängstigende Gefühle im Umgang mit diesem kennen sie nicht. Die Einschätzung ihrer Kompetenzen ist sehr selbstbewusst. Sie nimmt teilweise etwas ab. Dies könnte an den anspruchsvollen Inhalten des Unterrichts liegen. Das Spielen bleibt die beliebteste Beschäftigung. Allein der Glaube daran, dass der Computer keine Fehler mache, festigt sich hartnäckig, obwohl selbst fehlerhafte Programme erstellt werden. Die Einschätzung könnte möglicherweise auch auf eine fehlende Differenzierung zwischen der Bedienung von Computern und dem Schreiben von Programmen zurückzuführen sein.

Das Konzept zeigt, dass permanenter Computereinsatz im Fachunterricht auch mit dem Ziel möglich ist, informatische Grundkonzepte zu vermitteln. Basis des Erfolgs ist der kontinuierliche, über einen längeren Zeitraum hin stattfindende Unterricht, der die informatischen Grundkonzepte in systematischen Lernprozessen vermittelt. Die im Bildungsplan 2004 für Realschulen (Kultusministerium Baden-Württemberg, 2004b) beschriebenen informationstechnischen Kompetenzen und Inhalte und die Schlüsselqualifikationen entwickeln sich dann auf dem Fundament der informatischen Grundkonzepte bei der Arbeit ohne explizite Anwendungsschulungen.

7.2 Grenzen der Untersuchung

Einzelfallstudie

Das vorliegende Unterrichtskonzept wurde auf Basis einer Einzelfallstudie evaluiert. Die Auswahl der Versuchsklasse, deren Schülerinnen und Schüler aus der Grundschule ohne spezifische Vorkenntnisse bezüglich des Umgangs mit dem Computer kamen, und die vorliegenden schultypischen Rahmenbedingungen lassen eine bedingte Übertragung der Ergebnisse auf andere Klassen und Schulen zu. Da der Unterricht jedoch von der Forscherin selbst, einer Informatikerin, die keine Lehrbefähigung für die Schule und keinerlei Unterrichtserfahrung mit Schülerinnen und Schülern hatte, durchgeführt wurde, muss die Übertragbarkeit bezüglich der Lehrperson eingeschränkt werden. Im Gegensatz zur Forscherin fehlen den Fachlehrerinnen und Fachlehrern die zur Vermittlung der informatischen Grundkonzepte notwendigen fachlichen Kompetenzen. Sie müssten selbst eine umfassende informatische Basisausbildung erhalten, um den Schülerinnen und Schülern die verschiedensten grundlegenden Konzepte der Informatik bei der Interaktion mit Informatiksystemen deutlich machen zu können. Dies kann nicht in Anwendungsschulungen beim Schreiben von E-Mails, Surfen im Internet oder Verfassen von Texten in Textverarbeitungssystemen erfolgen.

Fehlender standardisierter Leistungstest für informatische Grundkonzepte

Der signifikante Unterschied der Leistungssteigerungen der Jungen und Mädchen vom Vor- zum Nachtest zu den informatischen Grundkonzepten hat nur unter Vorbehalt Gültigkeit, da aus Zeitgründen die Reliabilität und Validität des Tests nicht überprüft werden konnten (vgl. Kapitel 5.2.1). Das Erlernen der Konzepte ließe sich nur mit standardisierten Tests, die noch nicht existieren, empirisch valide evaluieren. Im Forschungsprojekt wurden zur Absicherung der Ergebnisse zusätzlich zahlreiche Unterrichtsmaterialien analysiert und ausgewertet, um die Ergebnisse abzusichern. Dies ist bei größeren Stichproben jedoch aus zeitlichen Gründen nicht möglich. Außerdem ist es schwierig, auf Basis der Auswertung schriftlicher Tests und Unterrichtsmaterialien informatische Strategien und Vorgehensweisen zu überprüfen, da nur Endprodukte von Arbeitsprozessen vorliegen. Zur Gewinnung von exakteren Ergebnissen müssten für die einzelnen informatischen Grundkonzepte stärker fokussierte Untersuchungsfragen gestellt und diese auf Basis von sehr präzisen Untersuchungsdesigns und Erhebungsinstrumenten analysiert werden.

Kurzfristige Ergebnisse

Durch die Anlage der Untersuchungskonzeption können keine langfristigen Auswirkungen gemessen werden. Die Englischlehrerin stellt zum Beispiel ein Jahr nach der Durchführung des Unterrichtsversuchs fest, dass das Leistungsniveau der Klasse sehr hoch sei. Auf den Englischunterricht bezogen ist ihrer Meinung nach der Anteil der Kinder, die die grammatikalischen Strukturen verstehen und anwenden können, sehr viel höher als in den 6. Klassen, die sie bis zu diesem Zeitpunkt unterrichtet hatte. Nur zwei bis drei Schülerinnen und Schüler hätten Schwierigkeiten. Ein Wirkungszusammenhang kann jedoch nicht nachgewiesen werden. Außerdem bleibt offen, inwiefern die informatischen Grundkonzepte das eigenständige Einarbeiten in neue Softwaresysteme unterstützen. Aus den Unterrichtsbeobachtungen kann abgeleitet werden, dass dies im Falle der in Imagine Logo integrierten Werkzeuge, wie zum Beispiel dem einfachen Texteditor, leicht möglich war.

7.3 Konsequenzen für die Implementierung des Unterrichtskonzepts

Übertragung des Unterrichtskonzepts auf die Klassenstufen 6 bis 10

Wie in Kapitel 7.1 beschrieben, war das entwickelte Unterrichtskonzept sehr erfolgreich. Informatische Grundkonzepte, Fachinhalte und ganz allgemeine Schlüsselkompetenzen wurden von den Schülerinnen und Schülern der Klassenstufe 5 erworben. Im Rahmen einer Grundbildung müssen die informatischen Grundkonzepte aber auch in höheren Klassenstufen wieder aufgegriffen und vertieft werden. Außerdem ist es notwendig, weitere grundlegende informatische Denk- und Arbeitsweisen in den Katalog der informatischen Grundkonzepte aufzunehmen. Das Unterrichtskonzept muss also auf die Klassenstufen

6 bis 10 übertragen werden. Dies bedeutet, entsprechende informatische Grundkonzepte für die verschiedenen Klassen auszuwählen und den relevanten Unterrichtsthemen aus einem erweiterten Fächerkatalog zuzuordnen. Wichtig ist, dass das bereits erworbene begriffliche informatische Wissen immer wieder aufgegriffen und spiralförmig weiter aufgebaut wird.

Für einige Fachinhalte werden in der Literatur Unterrichtsbeispiele mit Logo beschrieben. In den Berichten zur *Informatik und Datenverarbeitung in der Schule* der Pädagogischen Hochschule Ludwigsburg (vgl. Espenschied und Mauch 1984, Quehl 1985, Löthe & al. 1987, Weber 1988) werden vor allem Unterrichtseinheiten aus der Mathematik vorgestellt. Auch in aktuellen Logo-Veröffentlichungen sind Themen der Mathematik zu finden (vgl. Boytchev 2003, Farkas 2003). Um naturwissenschaftliches Arbeiten zu fördern, wird das Programmieren von Logo mit Lego-Aktivitäten verknüpft. Es entstehen dabei spielzeugähnliche Objekte (Alimisis & al., 2005) oder auch technische Lösungen für Alltagsprobleme (Sipitakiat und Cavallo, 2003). In den letzten Jahren wurden Logo-Versionen wie StarLogo¹ und NetLogo² entwickelt, in denen Tausende von Igel, ausgestattet mit Sensoren, parallel bewegt werden können. Mit diesen können dynamische Systeme, wie zum Beispiel die Ausbreitung von Krankheiten (Colella & al., 2001, S. 114), Stauentwicklungen (Resnick, 1996, S. 264) oder Bakterienkulturen (Resnick, 2001, S. 50), modelliert und simuliert werden.

Zusätzlich müssen weitere Software-Produkte hinsichtlich ihrer Eignung zur Veranschaulichung informatischer Grundkonzepte evaluiert werden. Dies schließt Standardanwendungen nicht aus. Wichtig ist dann die Entwicklung von Ideen und Konzepten, in denen die Anwendungen auf Basis der mentalen informatischen Modelle eingeführt werden. Einen Ansatz in diese Richtung beschreibt Bescherer (2005) mit den *LoDiCs* (Learning on Demand in Computing). Dabei handelt es sich um eine strukturierte Methode, die einen Rahmen vorgibt, wie die Vermittlung fundamentaler Konzepte der Informationstechnologie in beliebige Aufgabenstellungen integriert werden kann.

Aus- und Fortbildungskonzept für Lehrerinnen und Lehrer

Am dringlichsten ist sicherlich die Entwicklung eines Aus- und Fortbildungskonzepts für Lehrerinnen und Lehrer, das auf informatisches Grundlagenwissen und nicht auf Anwendungsschulungen ausgerichtet ist. Schon 1974 forderten Eyferth & al. (1974, S. 282):

„Mit der Aus- und Weiterbildung von Lehrern in Datenverarbeitung/Informatik muß sofort begonnen werden. Sie ist primär zu fördern.“

Ein zukünftiges Tätigkeitsfeld für die Forschung muss deshalb das Erarbeiten eines Konzepts für Lehrerinnen und Lehrer aller Fachbereiche sein, das realistisch umgesetzt werden kann. Hier können internationale Entwicklungen als Vorbild dienen und dem deut-

¹<http://education.mit.edu/starlogo> (Abruf: 03.08.05)

²<http://ccl.northwestern.edu/netlogo> (Abruf: 03.08.05)

schen Kontext angepasst werden. Im Zeitraum von 1982 bis 1988 wurden zum Beispiel in Parma in Italien mehr als 1000 Lehrerinnen und Lehrer in Logo fortgebildet. Sie erhielten eine Einführung in die Logo-Programmierung und die Methodik zur Umsetzung der erlernten Inhalte in ihrem Unterricht (Ambrosini & al., 1989). Ein Fortbildungsprogramm in Brasilien wählte einen anderen Ansatz. Dort wurde in den folgenden fünf Modulen das eigene Erlernen von Logo sehr stark mit dem Unterrichten vernetzt:

- Einführung in Logo
- Programmierung in Logo mit einem Kind
- Vertiefung der Logokenntnisse
- Programmierung in Logo mit einer ganzen Klasse
- Ausarbeitung eines pädagogischen Projekts.

Die Arbeiten mit den Schülerinnen und Schülern in Modul 2 und 4 wurden von dem Kursverantwortlichen beobachtet und anschließend mit den Lehrerinnen und Lehrern gemeinsam analysiert. Die Kommunikation erfolgte teilweise über das Internet (Valente, 1997).

Da nach dem neuen Bildungsplan 2004/2005 jede Lehrerin und jeder Lehrer an Realschulen in Baden-Württemberg die *Informationstechnische Grundbildung* unterrichten muss, ist der Fortbildungsbedarf enorm. Eine Umsetzung des Konzepts in der Breite ist nur dann möglich, wenn es in jeder Realschule zumindest einen Experten im Bereich der informatischen Grundkonzepte und Logo gibt. Bei 467 öffentlichen und privaten Realschulen³ (Stand Oktober 2004) ist dies allein schon ein großer Kraftakt. Eine Weiterbildung der Experten sollte nach dem brasilianischen Vorbild aus vernetzten Präsenz-, Selbstlern- und Transferphasen bestehen. In den Präsenz- und Selbstlernphasen muss der Wissenserwerb der informatischen Grundkonzepte und von Logo im Vordergrund stehen, die in Übungen am Rechner angewendet und vertieft werden. Die Betreuung der Übungen muss in den Präsenzveranstaltungen im Rechnerraum und in den Selbstlernphasen durch einen Online-Tutor erfolgen. Schon frühzeitig müssen konkrete Umsetzungen von Unterrichtssequenzen, in denen die informatischen Grundkonzepte in Fachinhalte eingebettet werden, mit den Kursverantwortlichen gemeinsam geplant und in den Transferphasen in den Schulen umgesetzt, ausgewertet und möglicherweise angepasst werden. Die Beschreibungen der Unterrichtsideen und die erforderlichen Materialien wie Arbeitsblätter oder Logo-Mikrowelten müssen den anderen Experten in einem gemeinsamen internetbasierten Arbeitsbereich wie zum Beispiel auf einem BSCW-Server⁴ zur Verfügung gestellt werden. Dadurch kann im Laufe der Fortbildung eine Sammlung guter Unterrichtsbeispiele entstehen, auf die alle zurückgreifen können. Eine Basisausbildung muss sich mindestens über ein Jahr erstrecken. Wie auch bei der Nachqualifikation der

³<http://www.statistik.baden-wuerttemberg.de/BildungKultur/Landesdaten/abschulen.asp> (Abruf: 03.08.05)

⁴Basic Support for Cooperative Work, <http://www.bscw.de> (Abruf: 03.08.05)

Lehrkräfte für Informatik an Gymnasien in Bayern im Programm SIGNAL⁵ müssen die Teilnehmer Anrechnungsstunden erhalten, um die höhere Arbeitsbelastung bewältigen zu können.

Nach der Fortbildung müssen die Experten in den Realschulen die Rolle der Multiplikatoren für die rund 16000 Kollegen⁶ (Stand Schuljahr 2004/2005) übernehmen. Sie geben im laufenden Schulbetrieb ihr Wissen über die informatischen Grundkonzepte und Logo in wöchentlichen Sitzungen anhand konkreter Unterrichtsbeispiele weiter. Außerdem stellen sie Unterrichtsmaterialien und didaktisch-methodische Überlegungen vor und sind permanenter Ansprechpartner.

Ein wichtiger Forschungsbereich in diesem Zusammenhang wird die Entwicklung aufeinander abgestimmter Konzepte und Materialien sein, die die Lehrerinnen und Lehrer sowohl als Lernende als auch Lehrende optimal unterstützen. Dies kann ähnlich zu der in Kapitel 3.5 beschriebenen Logo-Fortbildung erfolgen, in der die Lehrerinnen an ganz konkreten Unterrichtsbeispielen arbeiteten, die informatischen Grundkonzepte jedoch auf einem höheren Abstraktionsniveau thematisiert wurden. So kann das erworbene fachfremde Wissen schon kurze Zeit später an die Schülerinnen und Schüler weitergegeben werden.

⁵SIGNAL: Sofortprogramm Informatik am Gymnasium – Nachqualifikation von Lehrkräften, <http://www.stmwfk.bayern.de/hochschule/signal.html> (Abruf: 03.08.05)

⁶<http://www.statistik.baden-wuerttemberg.de/BildungKultur/Landesdaten/ablehrer.asp> (Abruf: 03.08.05)

Anhang A Empfehlungen zu einer informatischen Bildung

1976 **Zielsetzungen und Inhalte des Informatikunterrichts** (GI, 1976)

Schwerpunkte:

- Algorithmen finden und als Programm formulieren
- Praxisorientierte Probleme mit Informatiksystemen lösen
- Gesellschaftliche Auswirkungen der Datenverarbeitung kennen
- Theoretische und technische Grundlagen der Informatik erarbeiten

Bezeichnung: Informatik

Klassenstufe: —

Angebot: Pflichtbereich

1981 **Empfehlungen zur Einbeziehung informatischer Inhalte in den Mathematikunterricht der Sekundarstufe I** (GDM, 1981)

Schwerpunkte:

- Algorithmen formulieren und implementieren
- Strukturieren von Daten und Umgang mit großen Datenmengen

Bezeichnung: Informatik im Mathematikunterricht

Klassenstufe: Sek. I

Angebot: —

1985 **Empfehlungen und Überlegungen zur Gestaltung von Lehrplänen für den Computer-Einsatz im Unterricht der allgemeinbildenden Schulen** (MNU, 1985a)

Schwerpunkte:

- Anwendungen nutzen
- Problemlösungen durchführen
- Funktionsprinzipien von Geräten kennen und bedienen
- Gesellschaftliche Auswirkungen der Informationstechnologien kennen

Bezeichnung: Informationstechnische Grundbildung

Klassenstufe: nicht vor dem 8. Schuljahr

Angebot: wenige, komplexe Unterrichtseinheiten

1985 **Empfehlungen zur Gestaltung von Lehrplänen für die informationstechnische Bildung in der Sekundarstufe I. bzw. II und für den Computer-Einsatz im Mathematikunterricht der Sekundarstufe II** (MNU, 1985b)

Schwerpunkte:

- Anwendungen nutzen und anpassen
- Phasen der Problemlösung und Modellbildung durchführen
- Funktionsprinzipien von Soft- und Hardwaresystemen kennen und bedienen

- Gesellschaftliche Auswirkungen der Informationstechnologien kennen
 - Bezeichnung:* Informationstechnische Bildung
 - Klassenstufe:* im Anschluss an die Informationstechnische Grundbildung
 - Angebot:* Fakultativ oder Wahlpflichtfach
- 1986 **Rahmenempfehlung für die Informatik im Unterricht der Sekundarstufe I** (GI, 1986)
- Schwerpunkte:*
- Lösungsverfahren verstehen
 - Lösungen in Programmen formulieren
 - Arbeitsweisen des Computers kennen
 - Auswirkungen der Informationstechnik auf die Gesellschaft kennen
- Bezeichnung:* Informatik
- Klassenstufe:* Sek. I
- Angebot:* Pflichtbereich
- 1987 **Gesamtkonzept für die informationstechnische Bildung – Informationstechnische Grundbildung** (BLK, 1987)
- Schwerpunkte:*
- Individuelle Erfahrungen mit Informationstechniken aufarbeiten
 - Grundstrukturen und Grundbegriffe der Informationstechniken kennen
 - Computer und Peripherie bedienen
 - Problemlösungen in algorithmischer Form darstellen
 - Entwicklung der elektronischen Datenverarbeitung kennen
 - Chancen und Risiken der Informationstechniken einschätzen
- Bezeichnung:* Informationstechnische Grundbildung
- Klassenstufe:* Sek. I, jedoch nicht zu Beginn
- Angebot:* Einbettung in vorhandene Fächer
- 1987 **Gesamtkonzept für die informationstechnische Bildung – Vertiefende informationstechnische Grundbildung in Form der Informatik** (BLK, 1987)
- Schwerpunkte:*
- Funktionsprinzipien und Grenzen von Computern kennen und einschätzen
 - Problemlösemethoden kennen
 - Programmiersprachen lernen und strukturiert programmieren
 - Computer als Werkzeug in anderen Fächern einsetzen
 - Prozeßsteuerung durch Mikroprozessoren verstehen
- Bezeichnung:* Informatik
- Klassenstufe:* Sek. I (Wahlpflicht), Sek. II
- Angebot:* —

1993 GI-Empfehlungen für das Fach Informatik in der Sekundarstufe II allgemeinbildender Schulen (GI, 1993)

Schwerpunkte:

- Modellierungstechniken und ihre Darstellung kennen
- Theoretische Grundlagen von Computersystemen kennen
- Funktionsprinzipien und Grenzen von Computern verstehen
- Auswirkungen der Arbeit in vernetzten Systemen einschätzen

Bezeichnung: Informatik

Klassenstufe: Sek. II

Angebot: eigenes Fach

1995 Medienerziehung in der Schule – Orientierungsrahmen (BLK, 1995)

Schwerpunkte:

- Medien für verschiedene Aufgaben nutzen
- Wirkungsweise und Produktionsbedingungen von Medien kennen lernen
- Medien gestalten

Bezeichnung: Medienerziehung

Klassenstufe: 1 – 10

Angebot: —

1999 Informatische Bildung und Medienerziehung (GI, 1999)

Schwerpunkte:

- Daten automatisch verarbeiten
- Mit Informatiksystemen interagieren
- In Netzen kommunizieren

Bezeichnung: Informatische Bildung

Klassenstufe: Sek. I und II

Angebot: eigenes Fach Informatik

2000 Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen (GI, 2000)

Schwerpunkte:

- Mit Informatiksystemen interagieren
- Wirkprinzipien von Informatiksystemen kennen
- Informatische Modellierungstechniken anwenden
- Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft kennen

Bezeichnung: Informatische Bildung

Klassenstufe: 1 – 4, 5 – 13

Angebot: möglichst früh ein eigenes Fach Informatik

Anhang B Informatische Grundkonzepte – Vortest



Forschungsprojekt

Vermittlung informatischer Grundkonzepte im fächerübergreifenden Unterricht in Klasse 5 der Realschule

Test für Schülerinnen und Schüler (Vorerhebung)

Dipl. Inf. Birgit Wursthorn

- Informatische Grundkonzepte Teil 1

Institut für Mathematik und Informatik, Pädagogische Hochschule
Ludwigsburg

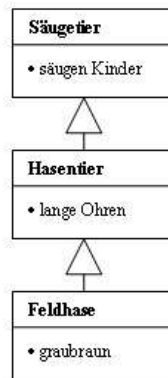
Als erstes musst du deinen Codenamen eintragen. Wie du dich vielleicht noch erinnern kannst, besteht dein Codename aus den ersten beiden Buchstaben des Vornamens der Mutter deiner Mutter (d.h. deiner Großmutter), gefolgt von dem Tag deiner Geburt und den ersten beiden Buchstaben des Vornamens deines Vaters.

Heißt die Mutter deiner Mutter "**Linda**", bist du am **12.09.1993** geboren und heißt dein Vater "**Manuel**", so lautet dein Code "**Li12Ma**". Schreibe jetzt deinen Codenamen in das dafür vorgesehene Feld.

Codename: _____

1) Die Lebewesen auf "Platurus"

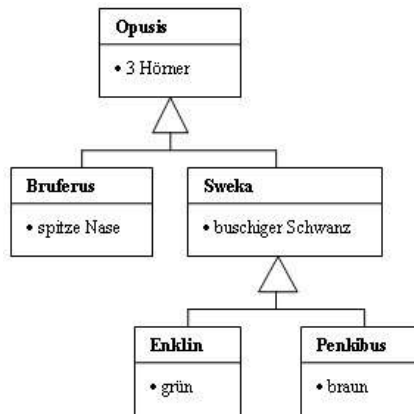
In der folgenden Darstellung sind die Eigenschaften der Tierarten "Säugetier", "Hasentier" und "Feldhase" und ihre verwandtschaftlichen Beziehungen abgebildet.



Folgende Informationen kann man aus der Darstellung links ablesen:

- Feldhasen sind graubraun, haben lange Ohren und säugen ihre Kinder.
- Hasentiere haben lange Ohren und säugen ihre Kinder
- Säugetiere säugen ihre Kinder

In "Platurus" leben ebenfalls Tierarten. Sie und ihre verwandschaftlichen Beziehungen sind im folgenden dargestellt:



Schreibe für die Tierarten "Opusis", "Sweka", "Penkibus" und "Bruferus" alle Eigenschaften auf, die man aus der Darstellung ablesen kann.

- Opusis: _____

- Sweka: _____

- Penkibus: _____

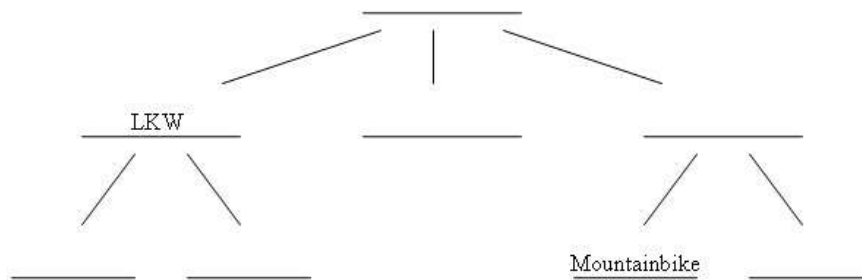
- Bruferus: _____

2) Einordnung von Begriffen

Ordne die folgenden Begriffe in das vorgegebene Schema ein.

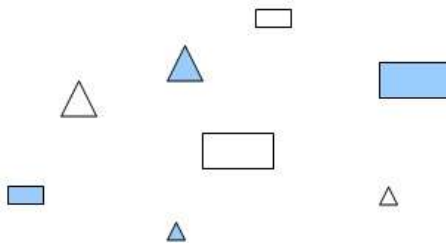
Fahrrad	Fahrzeug	Tanklaster
PKW	Kipplaster	Rennrad

Du musst die einzelnen Begriffe auf die freien Linien schreiben. 2 Begriffe wurden bereits in das Schema eingetragen.

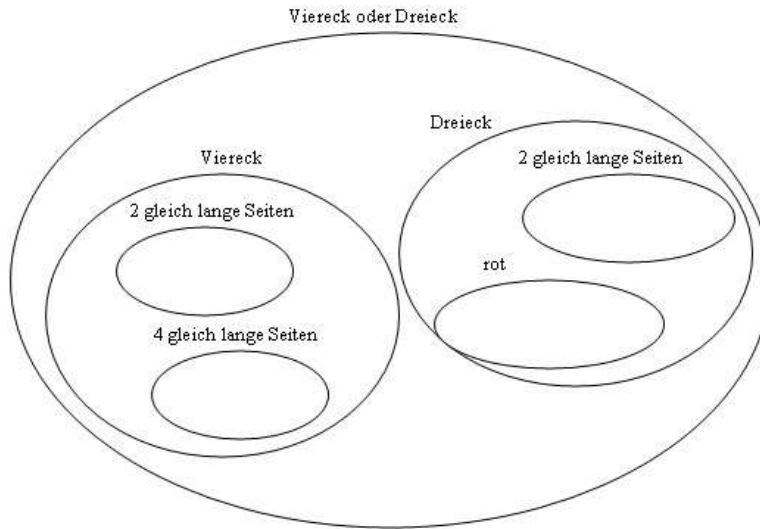


3) Ordnung von Figuren

Ordne die folgenden Figuren. Zeichne verwandte Figuren möglichst nahe zusammen.



4) Zeichne in das Schaubild mindestens 6 verschiedene Dreiecke oder Vierecke.



5) Folgen

Trage die nächsten 3 Zahlen auf den Linien ein.

4 8 12 16 _____

3 6 12 24 _____

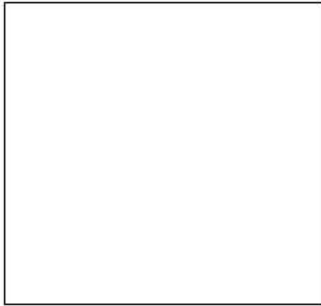
123 123 123 123 _____

Zeichne die nächsten 3 Symbole.



6) "Quadrekel"

"Quadrekel" das folgende Quadrat.



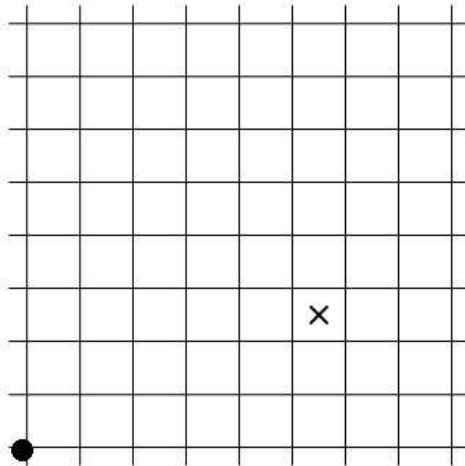
"Quadrekeln" bedeutet, ein Quadrat in 4 gleich große Quadrate zu unterteilen und das rechts oben entstandene Quadrat zu "Quadrekeln".

7) Zeichnung

Die vielen Quadrate unten sollen genau 1 cm breit und 1 cm hoch sein.

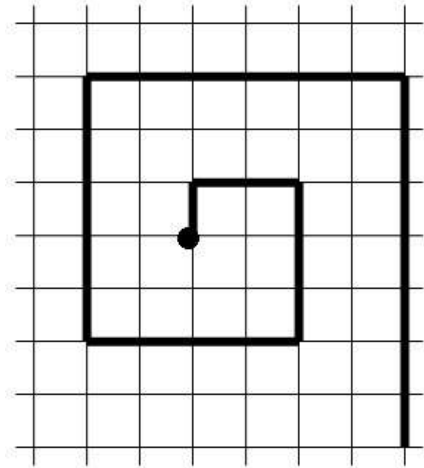
Führe die folgende Beschreibung ganz genau aus.

1. Starte die Zeichnung an dem kleinen Kreis.
2. Zeichne ein Quadrat mit der Seitenlänge 1 cm.
3. **Wiederhole** die folgenden beiden Anweisungen **bis das kleine Kreuz in der Zeichnung liegt**.
 - Starte die Zeichnung an dem kleinen Kreis.
 - Zeichne ein **um 1 cm größeres** Quadrat



8) Beschreibung

Beschreibe, wie die folgende Figur gezeichnet werden kann. Die Zeichnung muss an dem kleinen Kreis anfangen.





Forschungsprojekt

Vermittlung informatischer Grundkonzepte im fächerübergreifenden Unterricht in Klasse 5 der Realschule

Test für Schülerinnen und Schüler (Vorerhebung)

- Informatische Grundkonzepte Teil 2

Dipl. Inf. Birgit Wursthorn

Institut für Mathematik und Informatik, Pädagogische Hochschule
Ludwigsburg

Als erstes musst du deinen Codenamen eintragen. Wie du dich vielleicht noch erinnern kannst, besteht dein Codename aus den ersten beiden Buchstaben des Vornamens der Mutter deiner Mutter (d.h. deiner Großmutter), gefolgt von dem Tag deiner Geburt und den ersten beiden Buchstaben des Vornamens deines Vaters.

Heißt die Mutter deiner Mutter "**Linda**", bist du am **12.09.1993** geboren und heißt dein Vater "**Manuel**", so lautet dein Code "**Li12Ma**". Schreibe jetzt deinen Codenamen in das dafür vorgesehene Feld.

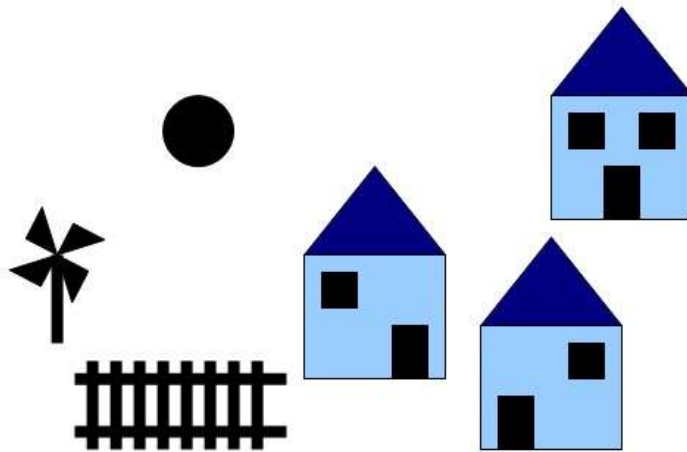
Codename: _____

1) Bilder

1. Beschreibe, wie 3 Personen zusammen das folgende Bild am schnellsten mit Pinsel und Malkasten malen können.



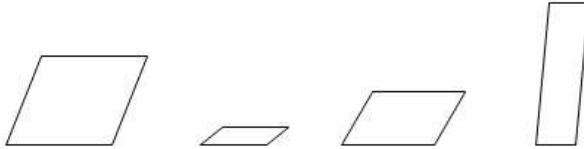
2. Das folgende Bild wurde mit Kartoffelstempeln erstellt. Beschreibe genau, wie 8 Personen das Bild möglichst schnell herstellen können. Es müssen die Kartoffelstempel geschnitzt und das Bild mit den Stempeln gedruckt werden.



3. Stell dir vor, du kannst einen Wunderstempel herstellen, dessen Größe verändert werden kann. Mit dem Wunderstempel



kannst du z.B. die Figuren



stempeln. Insgesamt kannst du natürlich noch viel mehr Figuren damit stempeln.

Zeichne bitte alle Wunderstempel auf, die du zur Herstellung des Stempelbildes auf der vorherigen Seite brauchst.

2) Adressliste

1. Schreibe in die folgende Zeile deine komplette Adresse.

2. Ordne möglichst übersichtlich 3 erfundene Adressen deiner Mitschülerinnen oder Mitschüler an, so dass du **aus der Anordnung** die Fragen und Aufgaben

- Wie viele Schülerinnen und Schüler kommen aus Östringen?
- Gib die Vornamen der Schülerinnen und Schüler an.
- Aus wie viel verschiedenen Ortschaften kommen die Schülerinnen und Schüler?
- Welcher Schüler wohnt in der Bahnhofstraße?

leicht ablesen und beantworten kannst.

Ordne nun zunächst die 3 Adressen übersichtlich an.

3. Beantworte die Fragen aus deinen aufgeschriebenen Adressen in den dafür vorgesehenen freien Zeilen.

- Wie viele Schülerinnen und Schüler kommen aus Östringen?

- Gib die Vornamen der Schülerinnen und Schüler an.

- Aus wie viel verschiedenen Ortschaften kommen die Schülerinnen und Schüler?

- Welcher Schüler wohnt in der Bahnhofstraße?

4. Ist deine übersichtliche Anordnung der Adressen auch geeignet, alle Adressen der Schülerinnen und Schüler deiner Realschule aufzunehmen? Begründe deine Antwort in den folgenden Zeilen.

3) Ein Rezept


1. Um Pfannkuchen für **eine Person** zu machen, benötigt man folgende Zutaten:

- 50 gr. Mehl
- 1 Ei
- 100 ml Milch
- 25 gr. Butter

2. Pfannkuchen für **zwei Personen** entstehen aus

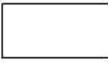

- 100 gr. Mehl
- 2 Eier
- 200 ml Milch
- 50 gr. Butter

3. Kannst du die Zutaten für das Rezept auch für 5 Personen in die folgenden Zeilen schreiben?

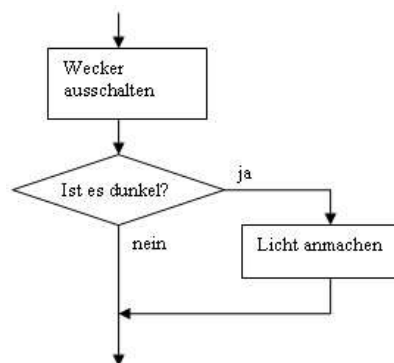
4. Schreibe als Letztes das Rezept nun mit Kreisen  anstatt mit Zahlen auf, so dass ich beim Ansehen deiner Arbeit eine von mir ausgedachte Zahl von Personen in die Kreise eintragen kann und das Rezept dann immer noch richtig ist.

4) Morgenqualen

Beschreibe in einem Diagramm, was du morgens zwischen dem Aufstehen und der Schule machst. Schreibe

- **Handlungen** in Rechtecke () und
- **Ja-Nein-Entscheidungen** in Karos ().

Damit dir die Lösung der Aufgabe leichter fällt, ist der Anfang vorgegeben. Vervollständige das Diagramm, auch wenn deine ersten Schritte etwas anders aussehen.





Forschungsprojekt

Vermittlung informatischer Grundkonzepte im fächerübergreifenden Unterricht in Klasse 5 der Realschule

Test für Schülerinnen und Schüler (Vorerhebung)

Dipl. Inf. Birgit Wursthorn

- Informatische Grundkonzepte Teil 3

Institut für Mathematik und Informatik, Pädagogische Hochschule
Ludwigsburg

Als erstes musst du deinen Codenamen eintragen. Wie du dich vielleicht noch erinnern kannst, besteht dein Codename aus den ersten beiden Buchstaben des Vornamens der Mutter deiner Mutter (d.h. deiner Großmutter), gefolgt von dem Tag deiner Geburt und den ersten beiden Buchstaben des Vornamens deines Vaters.

Heißt die Mutter deiner Mutter "**Linda**", bist du am **12.09.1993** geboren und heißt dein Vater "**Manuel**", so lautet dein Code "**Li12Ma**". Schreibe jetzt deinen Codenamen in das dafür vorgesehene Feld.

Codename: _____

1) Formen im Land "Droblavie"

Welche Formen existieren in dem Land "Droblavie", in dem es nur die Farben

- rot
- blau

und die Eigenschaften

- dreieckig
- viereckig

gibt. Zeichne alle Formen aus "Droblavie" auf.

2) "Formalien"

Die Sprache der Einwohner in "Formalien" hat nur die 3 Buchstaben



Ihre Wörter setzen sich aus **mehreren Buchstaben** zusammen. Sie dürfen jedoch **höchstens 2 gleiche Buchstaben** in einem Wort **nebeneinander** setzen.

In jeder Zeile steht ein Wort. Welche der folgenden Wörter haben die Einwohner von "Formalien" in ihrer Sprache richtig gebildet?



3) Roboter

Der Roboter "Dummie" kennt nur die folgenden beiden Wörter

- blub
- blob

Welche Ausrufe **mit höchstens 3 Wörtern** kann "Dummie" machen?

4) Geheimschrift

Du siehst unten eine Tabelle mit einer Geheimschrift

Buchstabe	Geheimcode
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

Löse die folgenden Aufgaben:

1. Finde das geheime Wort 72 65 85 83 und trage es in die folgende Zeile ein.

2. Schreibe das Wort BLUME in der Geheimschrift.

5) Deine Geheimschrift

1. Schreibe dein Geburtsdatum in die folgende Zeile.

2. Erfinde selbst eine Geheimschrift für die Darstellung eines Geburtsdatums und schreibe sie in den folgenden Kasten.

--

3. Gib dein Geburtsdatum in deiner Geheimschrift an.

4. Schreibe den 30.12.1786 in deiner Geheimschrift in die folgende Zeile.

6) Die 1. Steigerungsform von "schön", "hell" und "bunt"

Die Steigerung von **bunt** funktioniert folgendermaßen:

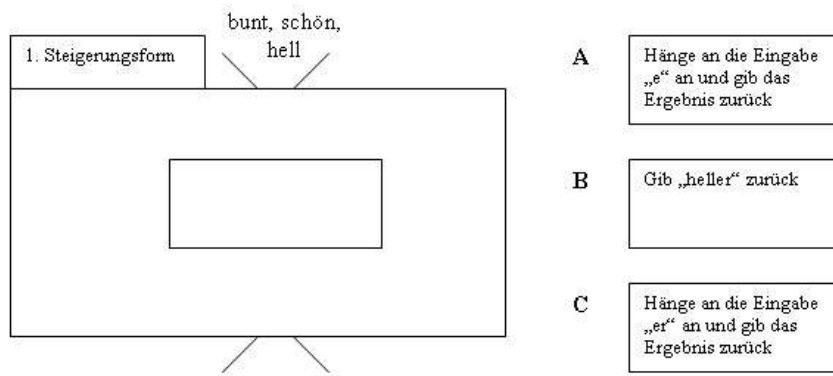
bunt	<u>bunter</u>	<u>am buntesten</u>
	1. Steigerungsform	2. Steigerungsform

Steigere du nun das Adjektiv **schön**.

schön	_____	_____
	1. Steigerungsform	2. Steigerungsform

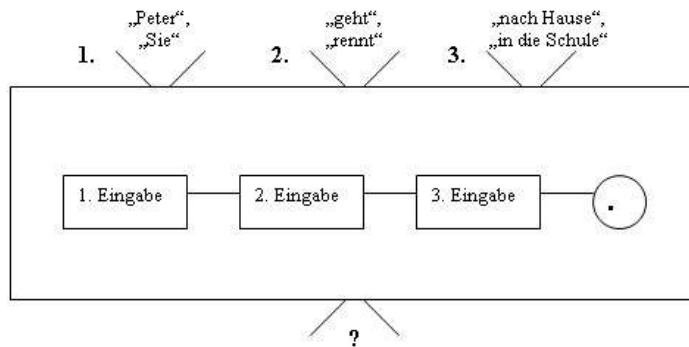
Auf der linken Seite siehst du eine Maschine. Diese Maschine kann die 1. Steigerungsform für die Adjektive "bunt", "schön" und "hell" erzeugen. Es fehlt jedoch der Motor der Maschine.

Mit welchem auf der rechten Seite dargestellten Motor kannst du die Maschine funktionstüchtig machen. Kreuze diesen Motor an.



7) Bildung von Sätzen

Die Maschine



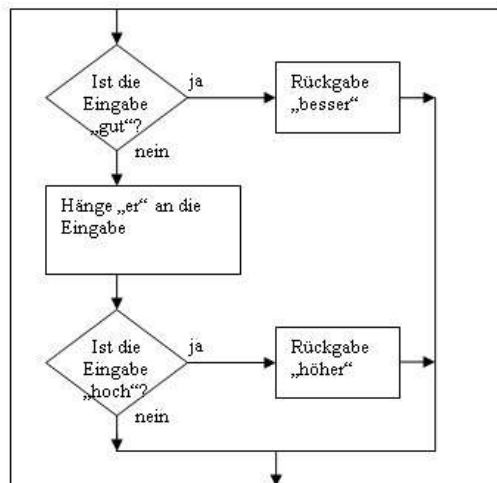
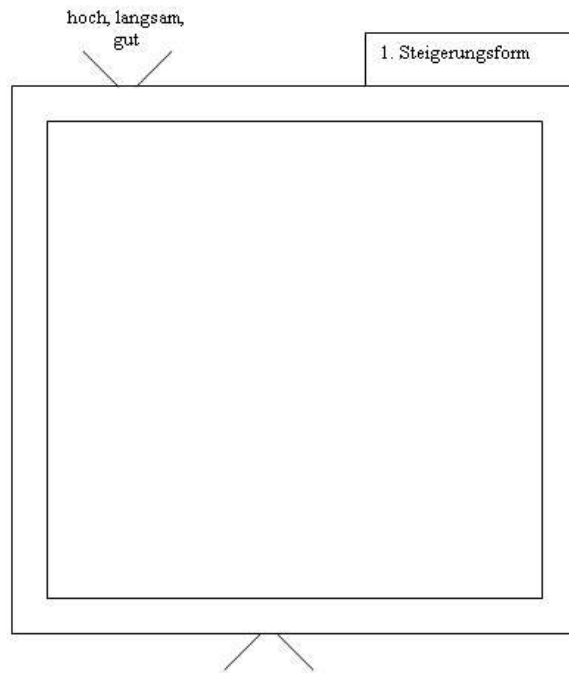
erzeugt gute Sätze.

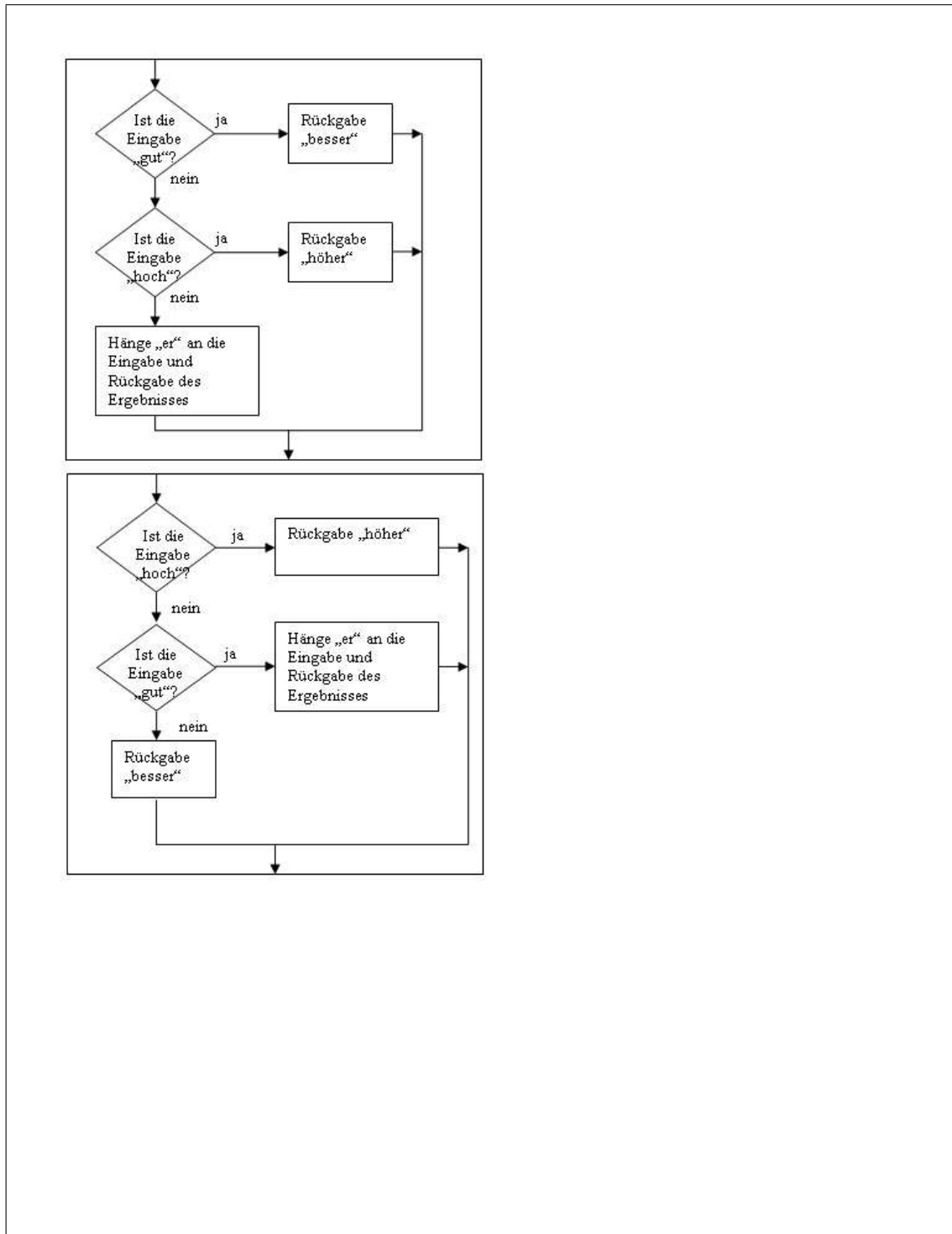
Welcher der folgenden Sätze wurden mit dieser Maschine gebaut? Kreuze diesen bitte an.

- ☐ Sie geht nach Hause.
- ☐ Peter rennt in die Schule
- ☐ Sie läuft nach Hause.
- ☐ Peter geht aus der Schule.

8) Die 1. Steigerungsform von "hoch", "langsam" und "gut"

Welcher Motor arbeitet hier korrekt. Kreuze ihn bitte an.





Anhang C Einstellungen zum Computer – Fragebogen

1) Motivationale Aspekte bei der Verwendung des Computers

		stimmt überhaupt nicht	stimmt eher nicht	teils teils	stimmt eher	stimmt voll
1.1)	Mir ist es wichtig, mit dem Computer zu arbeiten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.2)	Der Umgang mit dem Computer macht dumm.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3)	Ich finde Wissen über den Computer wichtig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.4)	Mir ist es wichtig, mit dem Computer zu spielen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.5)	Der Umgang mit dem Computer fordert mein Denken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.6)	Ich möchte mehr über den Computer wissen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.7)	Mir ist es wichtig, mit dem Computer zu lernen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.8)	Die Funktionsweise des Computers interessiert mich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.9)	Wissen über Computer bringt mir was.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2) Gefühle bei der Benutzung des Computers

	Welche Gefühle hast du bei der Arbeit mit dem Computer?	stimmt überhaupt nicht	stimmt eher nicht	teils teils	stimmt eher	stimmt voll
2.1)	Freude	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.2)	Langeweile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.3)	Aufregung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.4)	Herausforderung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.5)	Beunruhigung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.6)	Anregung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.7)	Bedrohung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.8)	Entspannung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.9)	Interesse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.10)	Angst	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.11)	Neugier	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.12)	Nervosität	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.13)	Spaß	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.14)	Unsicherheit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.15)	Gefühl der Zeitverschwendung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3) Selbsteinschätzung der Kompetenzen im Umgang mit dem Computer

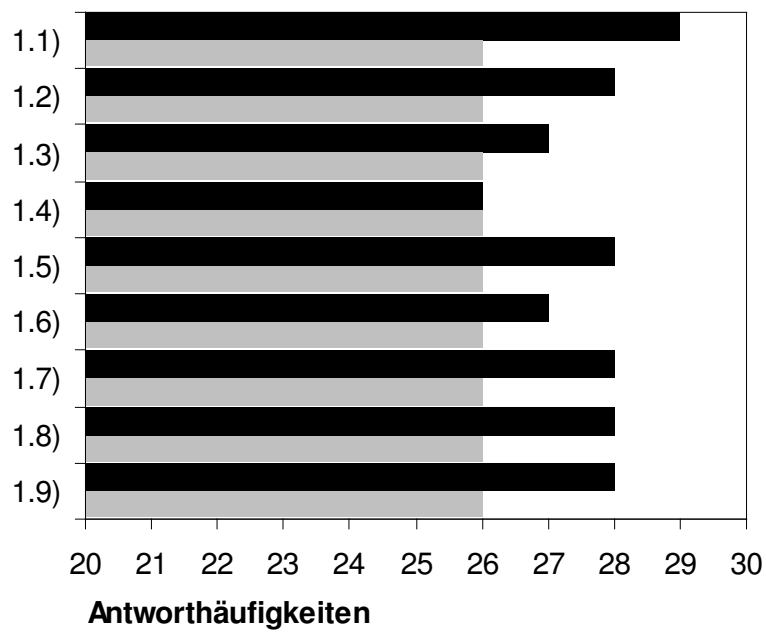
		stimmt überhaupt nicht	stimmt eher nicht	teils teils	stimmt eher	stimmt voll
3.1)	Ich traue mir im Umgang mit Computern viel zu.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.2)	Der Umgang mit dem Computer liegt mir nicht.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.3)	Den Umgang mit dem Computer werde ich nie richtig verstehen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.4)	Der Umgang mit Computern fällt mir nicht schwer.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.5)	Für die Nutzung des Computers habe ich einfach keine Begabung.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4) Beurteilung der Eigenschaften des Computers

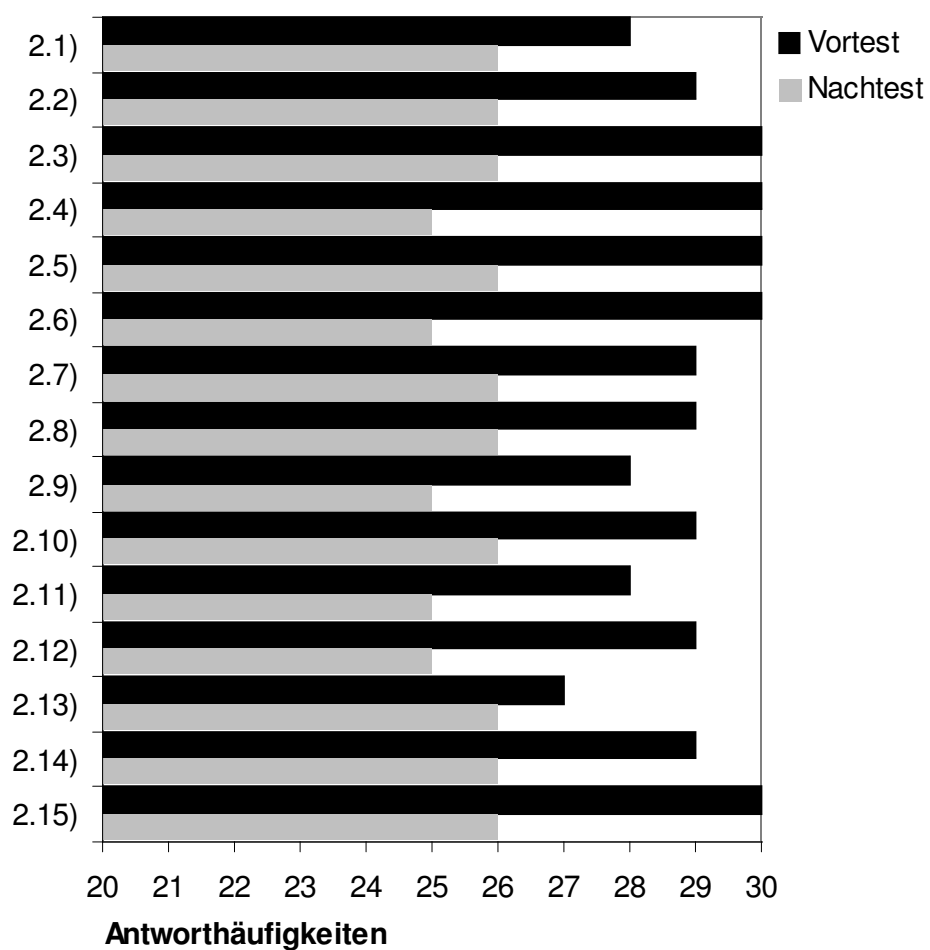
		stimmt überhaupt nicht	stimmt eher nicht	teils teils	stimmt eher	stimmt voll
4.1)	Der Computer kann ganz schnell rechnen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.2)	Der Computer kann mich verstehen, wenn ich Probleme habe.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.3)	Der Computer kann nur arbeiten, wenn ich ihm genau sage, was er tun soll.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.4)	Der Computer ist undurchschaubar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.5)	Der Computer erkennt selbst, was ich brauche und gibt es mir dann sofort.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.6)	Der Computer kann nur das arbeiten, was in dem benutzten Programm steht.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.7)	Der Computer ist böartig.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.8)	Der Computer macht nur Fehler, wenn er falsch bedient wird.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.9).	Der Computer kann alles.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.10)	Dadurch dass ich einen Computer benutzen kann, muss ich selbst nicht mehr so viel lernen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Anhang D Rücklauf der Fragebögen zu den Einstellungen zum Computer

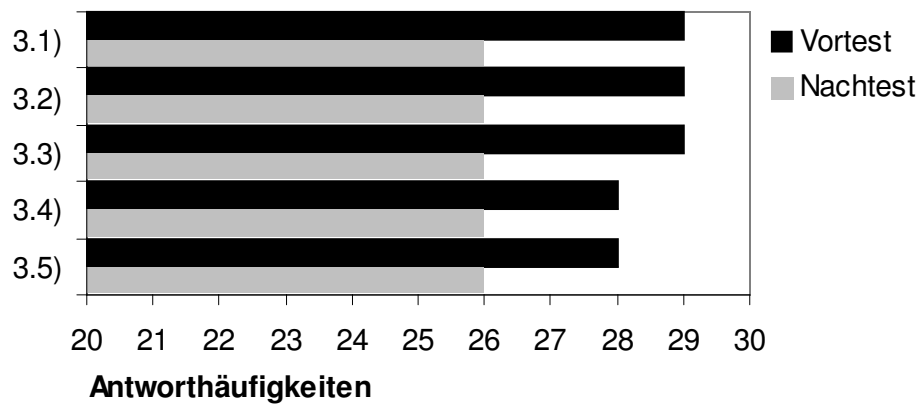
1) Motivationale Aspekte bei der Verwendung des Computers



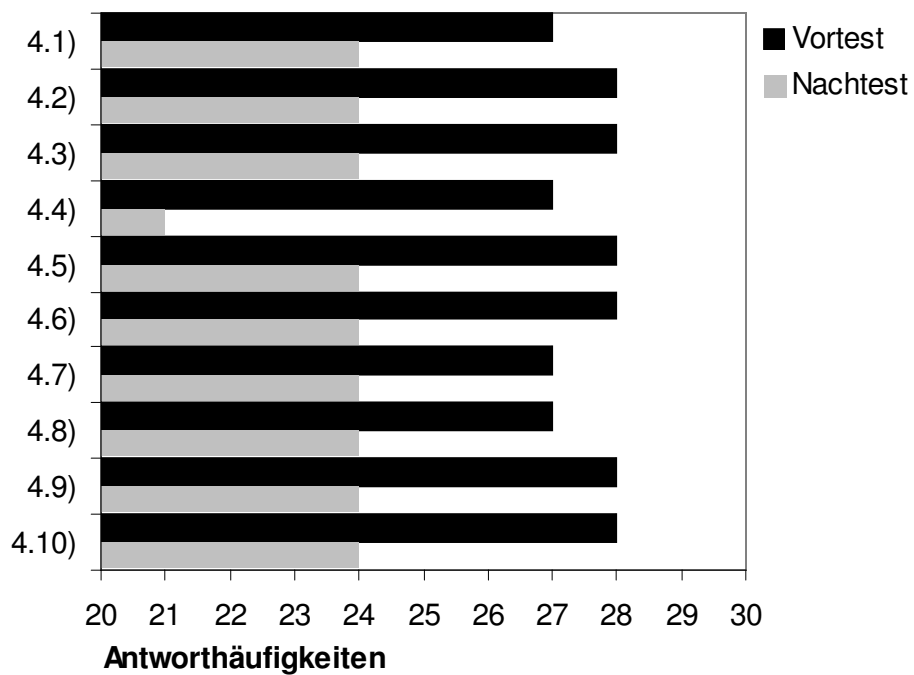
2) Gefühle bei der Benutzung des Computers



3) Selbsteinschätzung der Kompetenzen im Umgang mit dem Computer



4) Beurteilung der Eigenschaften des Computers



Anhang E Interviewleitfaden Lehrerinnen

Wir haben jetzt ein Jahr lang die Klasse bei der Vermittlung von informatischen Grundkonzepten in den Fächern Mathematik, Deutsch, Englisch und Musik betreut.

I) Zu den einzelnen Themen wurden von mir für die Schülerinnen und Schüler **Arbeitsblätter** erstellt.

1. Wie bewerten Sie die Arbeitsblätter für den Unterricht?
2. Es wurden vier verschiedene Formen von Arbeitsblättern verwendet:
 - zum selbstständigen Durcharbeiten
 - zum selbstständigen Durcharbeiten mit Punkten für einzelne Aufgaben
 - zum selbstständigen Erarbeiten ganz neuer Inhalte und gleichzeitigem bewerteten Test
 - zum gemeinsamen Durcharbeiten mit der LehrpersonWelche Formen haben sich bewährt?

II) Für viele Inhalte wurden den Schülerinnen und Schülern von mir **Programme**, so genannte Mikrowelten, vorgegeben.

1. Waren die Themen der Mikrowelten interessant oder eher künstlich für die Schülerinnen und Schüler?
 - a) Nennen Sie bitte Beispiele.
2. Waren die Programme offen genug, damit sie von den Schülerinnen und Schülern sinnvoll ausgebaut werden konnten?
 - a) **Falls ja**: An welche Programme denken Sie?
 - b) **Falls nein**: Welche Programme konnten von den Schülerinnen und Schülern nicht genügend ausgebaut werden?
 - c) **Falls nein**: Wie hätten diese Programme anders gestaltet werden müssen, damit sie die Schülerinnen und Schüler sinnvoll hätten einsetzen können?

III) Der Unterricht im Forschungsprojekt fand in ganz verschiedenen **Unterrichtsformen** statt.

1. Der Computerunterricht war häufig in Frontalphasen und in Gruppenarbeit am Computer aufgeteilt. Wie beurteilen Sie diese Aufteilung?
2. Der Computerunterricht fand für die Schülerinnen und Schüler nicht nur am Computer statt. Wie beurteilen Sie die Aufteilung zwischen Computerarbeitsphasen und Arbeitsphasen ohne Computer?
3. Solange wir die Computer im Computerraum verwendet haben, fand ein Teil des Computerunterrichts im Klassenzimmer statt. Für die Computerarbeit musste die Klasse dann in den Computerraum wechseln. Wie beurteilen Sie die Aufteilung zwischen der Arbeit im Klassenzimmer und im Computerraum?

IV) Ab März haben die Schülerinnen und Schüler über fünf Wochen an dem **Ritterprojekt** gearbeitet.

1. Sehen Sie das Ritterprojekt als erfolgreich an?
 - a) Wie beurteilen Sie das Projekt?
2. Wurde mit der Projektarbeit zum richtigen Zeitpunkt begonnen?
 - a) **Falls nein:** Welchen Zeitpunkt hätten Sie als geeignet angesehen?
 - b) **Falls nein:** Was wäre an diesem Zeitpunkt besser gewesen?
3. Haben Sie den Eindruck, dass den Schülerinnen und Schülern genügend Zeit zur Verfügung stand?
 - a) Für was wäre mehr/weniger Zeit notwendig gewesen?
4. Wie passend beurteilen Sie das Thema *Rittergeschichte* für eine Projektarbeit in Klasse 5?
5. Wie sind die Schülerinnen und Schüler mit den Freiheiten der Projektarbeit zu-rechtgekommen?
6. Beim Gedicht und beim Grundriss der Burg haben viele Schülerinnen und Schüler zum Beispiel nur die Pflichtvorgaben abgearbeitet. Sie haben weder versucht, das Gedicht zu erweitern noch einen interessanteren Grundriss der Burg zu konstruieren. Warum haben sie ihr Ritterprojekt in dieser Hinsicht nicht ausgebaut und dadurch interessanter gestaltet?
7. Im Ritterprojekt haben die Schülerinnen und Schüler weniger an konkreten Fachinhalten als im Unterricht zuvor gearbeitet. Ist nun zur Vermittlung der theoretischen informatischen Grundkonzepte der integrierte Fachunterricht oder Projektarbeit besser geeignet?
 - a) Was haben Sie während des Schuljahrs beobachtet?

V) Auch im Forschungsprojekt wurden den Schülerinnen und Schülern **Hausaufgaben** aufgegeben, die sich auf die Inhalte des Fachs oder auf das theoretische Üben der informatischen Grundkonzepte bezogen. Zu Beginn des Schuljahrs gab es im Computerunterricht regelmäßig Hausaufgaben, mit Beginn des Ritterprojekts fast keine mehr.

1. Wie beurteilen Sie die Hausaufgaben im Computerunterricht?

VI) Der Computerunterricht fand in Ihrem Fach **integriert** statt.

1. Hat der Computerunterricht auf den normalen Unterricht in Ihrem Fach bei den Schülerinnen und Schülern zurückgewirkt?
 - a) Was haben Sie beobachtet?
2. Hatten Sie Probleme, die Inhalte aus dem Lehrplan in der verkürzten Zeit zu vermitteln?
 - a) **Falls ja:** Lohnt es sich trotzdem, diesen Nachteil in Kauf zu nehmen?
3. Welche Verknüpfungen zwischen den Fachinhalten und den theoretischen informatischen Grundkonzepten waren für Ihr Fach geeignet/ungeeignet?
 - a) Können Sie Beispiele angeben?

4. Unterstellen wir einmal, dass das Forschungsprojekt wiederholt wird.
 - a) Sollten dann die Computerphasen im Fachunterricht ausgebaut oder eingeschränkt werden?
 - b) Können Sie konkrete Themenbereiche benennen, an denen Änderungen vorgenommen werden sollten?

VII) Der Computerunterricht im Forschungsprojekt war **fächerübergreifend** konzipiert.

1. Wurden die Inhalte fächerübergreifend vermittelt oder liefen die Stunden in den einzelnen Fächern nebeneinander her?
 - a) Können Sie Beispiele angeben?
2. Haben Inhalte aus anderen beteiligten Fächern in Ihrem Fach gewirkt?
 - a) **Falls ja:** Geben Sie Beispiele an.
3. Durch den fächerübergreifenden Aspekt wurden die theoretischen informatischen Grundkonzepte in verschiedenen Fächern angesprochen. Zum Beispiel wurde sowohl in Deutsch als auch in Englisch mit Syntaxdiagrammen gearbeitet. Wurden diese theoretischen informatischen Grundkonzepte häufig genug oder vielleicht auch zu oft wiederholt?

VIII) Jetzt noch ein paar Fragen zum **Lernerfolg** der Schülerinnen und Schüler im Computerunterricht.

1. Was haben die Schülerinnen und Schüler, ganz allgemein gesagt, im Computerunterricht in diesem Schuljahr überhaupt gelernt?
2. Welche Schülerinnen und Schüler haben von dem Computerunterricht profitiert?
 - a) Aus welchem Grund haben nur diese Schülerinnen und Schüler vom Computerunterricht profitiert?
3. Betrachtet man den Gesamtunterricht in den vier beteiligten Fächern zusammen mit dem Computerunterricht, welche Schülerinnen und Schüler hatten dann eher Nachteile?
 - a) Welche Nachteile waren das? Was konnten Sie genau beobachten?
4. Die Schülerinnen und Schüler haben das ganze Jahr im Computerunterricht durchgehalten. Was war ausschlaggebend dafür? (Computer AG, Eltern, ...)
5. Konnten Sie beobachten, dass die Schülerinnen und Schüler, die zusätzlich die Computer AG besuchten, einen Vorteil im normalen Computerunterricht hatten?
 - a) In welchen Bereichen hatten sie Vorteile?

IX) Der Computerunterricht wurde von den Schülerinnen und Schülern ganz unterschiedlich **angenommen**.

1. Können Sie die unterschiedlichen Motivationen der Schülerinnen und Schüler beschreiben?
2. Gab es bei einzelnen Schülerinnen und Schülern Unterschiede bezüglich der Moti-

vation im Computerunterricht und Ihrem Unterricht in den anderen Stunden des Fachs?

- a) Was haben Sie beobachtet?
3. Hat sich die Motivation der Schülerinnen und Schüler über das Schuljahr hinweg verändert?
 - a) Wann gab es Höhen und Tiefen?
 - b) Können Sie sich diese Höhen und Tiefen erklären?
4. Wie schätzen Sie die Ausdauer der Schülerinnen und Schüler im Computerunterricht ein?
5. Unterscheidet sich die Ausdauer der Schülerinnen und Schüler bei der Arbeit im Computerunterricht und in dem von ihnen gestalteten Fachunterricht?
 - a) **Falls ja:** Wie können Sie sich diese Unterschiede erklären?

X) Jetzt noch ein paar Fragen zu meiner **Person**.

1. Wie habe ich auf die Schülerinnen und Schüler gewirkt?
2. Wie habe ich mich im Laufe des Schuljahrs entwickelt?
3. Welche Ratschläge für die Zukunft würden Sie mir geben, wenn ich das Forschungsprojekt noch einmal durchführen würde?

XI) Einige **Rahmenbedingungen** im Forschungsprojekt waren entweder durch die Schule oder durch meine Person vorgegeben. An dieser Stelle möchte ich gerne ihre Meinung bezüglich dieser Rahmenbedingungen und ihrer Auswirkungen auf das Forschungsprojekt erfahren.

1. Sind Doppelstunden für den Computerunterricht geeignet? (Argumente)
2. Braucht jede Schülerin und jeder Schüler einen eigenen Computer? (Argumente)
3. Muss die Fachlehrerin/der Fachlehrer auch die Person sein, die den Computerunterricht durchführt? (Argumente)
4. Welche anderen Rahmenbedingungen fallen Ihnen noch ein, die irgendeinen Einfluss auf den Projektverlauf hatten? Benennen Sie diese zusammen mit ihrem Einfluss auf das Projekt.
5. Braucht man zur Veranschaulichung im Klassenzimmer auch Präsentationsmöglichkeiten wie einen Beamer? (Argumente)
6. Welche Auswirkung hatte die Tatsache, dass nur mit Logo über das gesamte Jahr gearbeitet wurde?

XII) Die Art des Computerunterrichts im Forschungsprojekt könnte von Ihnen im Rahmen des integrierten ITG-Unterrichts im neuen Bildungsplan in den nächsten Schuljahren fortgeführt werden.

1. Werden Sie diese Art von Unterricht fortführen?
 - a) **Falls ja:** Was werden Sie genau machen? Nennen Sie einige Beispiele?
 - b) **Falls nein:** Was hindert Sie daran, den Unterricht in dieser Art fortzusetzen?

2. Wie ist Ihr Eindruck nach einem Jahr der Vermittlung der informatischen Grundkonzepte. Ist die 5. Klassenstufe überhaupt geeignet, um diese zu vermitteln?
 - a) Welche Argumente sprechen dafür und welche dagegen?
3. Waren die Schülerinnen und Schüler generell gesehen mit dem Computerunterricht überfordert?
 - a) Bei welchen Inhalten oder wann ist Ihnen das besonders aufgefallen?

Gibt es weitere Punkte, die Ihnen rückblickend auf das Forschungsprojekt wichtig erscheinen?

Anhang F Interviewleitfaden Schülerinnen und Schüler

I) Wir unterhalten uns jetzt etwas über deine **Erfahrungen** im Computerunterricht in diesem Schuljahr.

1. Bist du gerne in den Computerunterricht gegangen?
2. Was hat dir im Computerunterricht besonders gut gefallen? Was war spannend?
3. Was hat dir im Computerunterricht nicht gefallen?
4. Hast du im Computerunterricht immer alles gleich verstanden?
 - a) **Falls nein**: Mit was kamst du überhaupt nicht klar?
5. Erzähl mal, was du im Computerunterricht am liebsten gemacht hast?
6. Was war im Computerunterricht so wichtig für dich, dass du es zu Hause gerne weiter machen möchtest?
7. Auf welche deiner Arbeiten im Computerunterricht warst du so richtig stolz?

II) In seinen Lieblingsfächern **arbeitet** man gerne im Unterricht **mit**.

1. Hast du im Computerunterricht gut mitgearbeitet?
 - a) **Falls ja**: Kannst du dich noch erinnern, bei welchen Themen du gut mitgearbeitet hast?
 - i. **Falls ja**: Welche Themen waren es?
 - ii. **Falls ja**: Woran lag es, dass du bei diesen Themen gut mitgearbeitet hast?
 - b) **Falls nein**: Warum hast du im Computerunterricht nicht gut mitgearbeitet?

III) Du hattest den **Computerunterricht** bei mir und die **anderen Stunden** in den Fächern Mathematik, Deutsch, Englisch und Musik bei deinen Fachlehrerinnen.

1. Fandest du den Computerunterricht schwerer als den normalen Unterricht in den Fächern Mathematik, Deutsch, Englisch und Musik?
 - a) **Falls ja**: Was war denn im Computerunterricht schwerer?
 - b) **Falls nein**: War der Computerunterricht dann leichter?
 - i. **Falls ja**: Was war denn im Computerunterricht leichter?
2. Was war im Computerunterricht eigentlich anders als im normalen Unterricht?
3. Für die Arbeit am Computer habe ich immer eine längere Zeit eingeplant, damit jedes Schülerpaar am Computer in seinem eigenen Tempo arbeiten konnte.
 - a) Konntest du in deinem eigenen Tempo arbeiten?
 - b) Wie fandest du das?

IV) Es war von mir geplant, dass ihr während der Arbeit am Computer euch viel intensiver mit euren Mitschülerinnen und Mitschülern **austauschen** könnt, indem ihr zusammen mit eurem Partner die Aufgaben am Computer löst und euch auch die Arbeiten eurer Mitschülerinnen und Mitschüler an den anderen Computern anschaut.

1. Hast du gerne mit einem Partner an einem Computer zusammengearbeitet oder hättest du lieber alleine gearbeitet?
2. Habt ihr am Computer alles zusammen gemacht oder die Aufgaben eher aufgeteilt?
3. Hast du Mitschülerinnen und Mitschülern an anderen Rechnern auch manchmal geholfen? Was zum Beispiel?
4. Haben dir andere auch geholfen? Was zum Beispiel?
5. Hast du bei Problemen eher die Lehrerin oder eher deine Mitschülerinnen und Mitschüler zu Hilfe gerufen?
6. Bist du während des Unterrichts auch zu den Rechnern anderer Mitschülerinnen und Mitschüler gegangen? Was hast du dort gemacht oder beobachtet?

V) Im Computerunterricht hast du ganz viele **Arbeitsblätter** bekommen. Hier siehst du noch einmal die Ordner der einzelnen Fächer und des Ritterprojekts mit den verschiedenen Arbeitsblättern.

1. Hast du die Arbeitsblätter gerne gelesen oder eher nicht?
 - a) **Falls ja**: Waren die Texte und Fragen auf den Arbeitsblättern zu schwer?
 - i. **Falls ja**: Was war zum Beispiel zu schwer?
 - b) **Falls nein**: Warum hast du die Arbeitsblätter zum Beispiel nicht gerne gelesen?
2. Wie hast du dir helfen lassen, wenn du die Aufgaben auf den Arbeitsblättern nicht gleich verstanden hast?

VI) Wir haben das ganze Schuljahr mit **Logo** gearbeitet. Zunächst haben wir eine deutsche (MSW Logo) und dann eine englische Version (Imagine Logo) verwendet.

1. Gab es insgesamt etwas, was dich bei der Arbeit mit Logo gestört hat?
2. Du hast ja auch schon mit anderen Programmen mit dem Computer gearbeitet. Mit welchen anderen Programmen hast du schon gearbeitet?
3. Vergleiche nun deine Arbeit mit Logo mit der Arbeit mit den anderen Programmen. Was war an der Arbeit mit Logo Besonders für dich?
4. Kannst du den Unterschied zwischen MSW Logo und Imagine Logo beschreiben?
5. Was hat dir an Imagine Logo besonders gut gefallen?

VII) Vor den Osterferien haben wir mit dem **Ritterprojekt** angefangen und etwa 5 Wochen daran gearbeitet. Hier kannst du dir am Rechner noch einmal deine Arbeit anschauen.

1. Was war während des Projektunterrichts anders als zuvor im Computerunterricht?
2. Was hat dir besonders gefallen?
3. Was fandest du nicht gut?
4. Hättest du gerne mehr Zeit für dein Ritterprojekt gehabt?
5. Was hättest du gerne noch zusätzlich eingebaut?
6. Warum hast du im Ritterprojekt beim Gedicht oder beim Zeichnen des Grundrisses

der Ritterburg nicht mehr ausprobiert?

VIII) Jetzt interessiert mich noch, was du im Computerunterricht **gelernt** hast.

1. Was hast du im Computerunterricht über den Computer und seine Bedienung gelernt? Erzähl mir ein paar Beispiele.
2. Hast du im Computerunterricht auch etwas über Mathematik, Deutsch, Englisch oder Musik gelernt? Was zum Beispiel?
3. Von den Dingen, die du im Computerunterricht gelernt hast, was fandest du besonders wichtig?
4. Hättest du dich im Computerunterricht mehr angestrengt, wenn es eine Note im Zeugnis gegeben hätte?

IX) Ein ganzes Schuljahr hat der Computerunterricht nun schon stattgefunden.


1. Bist du froh, dass der Computerunterricht vorbei ist oder nicht?
 - a) Kannst du mir das erklären?
2. Stell dir vor, ich mache im nächsten Schuljahr in deiner Klasse den Computerunterricht so wie in diesem Schuljahr weiter. Fändest du das gut?
 - a) Was sollte dann anders sein?
3. Hast du deinen Eltern über deine Arbeiten im Computerunterricht erzählt?
 - a) Was zum Beispiel?
4. Hast du deinen Freundinnen oder Freunden über deine Arbeiten im Computerunterricht erzählt?
 - a) Was zum Beispiel?

X) Jetzt habe ich dich ganz viele Dinge gefragt.

1. Möchtest du mir noch etwas zum Computerunterricht sagen, was dir ganz wichtig ist?

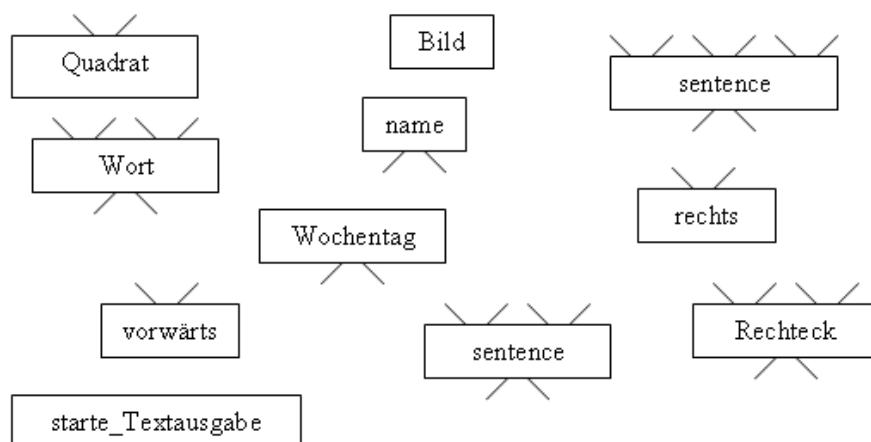
Anhang G Informatische Grundkonzepte – Tests im laufenden Schuljahr

Test 1: *Funktionen*

	Test „Funktionen“	Name: _____	Datum: _____	Seite 1
---	--------------------------	-------------	--------------	----------------

Aufgabe 1


Welche der folgenden Maschinen sind Funktionen?



Aufgabe 2

1. Schreibe in die folgenden Zeilen die Funktion *town*, die zufällig eine der englischen Städte Chester, London oder Ashton zurückgibt.

2. Male eine Maschine für die Funktion *town*.

	Test „Funktionen“	Name:	Datum:	Seite 2
---	--------------------------	-------	--------	----------------

Aufgabe 3

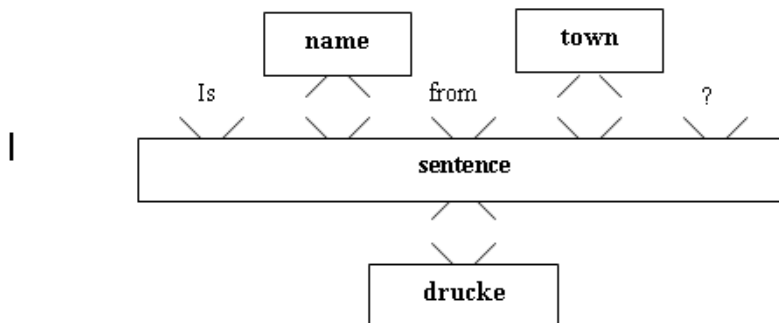
- Die Funktion *name* gibt einen der Namen Ben, Nick oder Debbie zurück. Sie sieht folgendermaßen aus:

pr name

rg zel [Ben Nick Debbie]

ende

- Schreibe 2 Fragen auf, die von der folgenden Maschine erzeugt werden können und beantworte diese dann auf Englisch.




1. Frage: _____

Antwort: _____

2. Frage: _____

Antwort: _____

- Schreibe in die folgende Zeile, was du in der Befehlszeile in Logo eingeben musst, damit die englischen Fragen im Grafikfenster ausgegeben werden.

	Test „Funktionen“	Name:	Datum:	Seite 3
---	--------------------------	-------	--------	----------------


Aufgabe 4

1. Male eine Maschine, die die folgenden Fragen erzeugt und dann im Grafikfenster ausgibt.

Is Nick the brother of Debbie ?
Is Ben the brother of Nick ?
Is Debbie the brother of Ben ?

2. Schreibe in die folgende Zeile, was du in der Befehlszeile in Logo eingeben musst, damit folgende englischen Fragen im Grafikfenster ausgegeben werden.

Is this Nick ?
Is this Ben ?
Is this Debbie ?

	Test „Funktionen“	Name:	Datum:	Seite 4
---	--------------------------	-------	--------	----------------


Zusatzaufgabe

1. Logo soll folgende Fragen im Grafikfenster ausgeben:

Is the desk green ?
Is the chair red ?
Is the pencil blue ?
Is the school bag yellow ?

2. Schreibe auf diese Seite die neuen Funktionen, die du für diese Fragen brauchst.
3. Male darunter die gesamte Maschine, die die Fragen erzeugt.
4. Schreibe dann auf, was du in der Befehlszeile eingeben musst, damit die Fragen im Grafikfenster ausgegeben werden.

Test 2: Syntaxdiagramme, Funktionen und Programmablaufpläne

	Test „Syntaxdiagramm“, „Maschinen“	Name:	Datum:	Seite 1
---	---	-------	--------	----------------

Aufgabe 1 (12 Punkte)

1. Zeichne ein Syntaxdiagramm, das die beiden Sätze

Die Tulpe wächst in Omas Garten ,
Blume **Ort**


Die Mohnblume wächst auf dem Feld .
Blume **Ort**

beschreibt. (4 Punkte)

2. Schreibe in die 2 freien Zeilen zwei neue Sätze, die nach dem Syntaxdiagramm gebildet werden. (2 Punkte)

3. Zeichne für die beiden Funktionen *Blume* und *Ort* Maschinen. (2 Punkte)

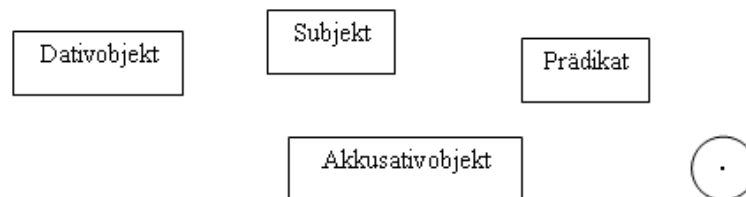
4. Zeichne die gesamte Maschine, mit der die Sätze aus Aufgabe 1.1 erzeugt werden können. Aus einzelnen Wörtern erhält man mit der Funktion *Satz* (englisch *sentence*) einen ganzen Satz. (4 Punkte)

	Test „Syntaxdiagramm“, „Maschinen“, „Funktionen“	Name:	Seite 2
---	---	-------	----------------

Aufgabe 2 (3,5 Punkte)

Ich schreibe der Großmutter einen Brief.

Du gibst Peter ein Geschenk.



1. Zeichne ein Syntaxdiagramm, das die Satzglieder für die beiden Sätze in die richtige Reihenfolge bringt. Verwende die vorgegebenen Bausteine. (2,5 Punkte)
2. Schreibe in die freie Zeile einen weiteren Satz, der mit Hilfe des Syntaxdiagramms gebildet werden kann. (1 Punkt)

Aufgabe 3 (4 Punkte)


Hier siehst du die Funktion *Befehlsform* in Logo (links) und als Maschine (rechts).

```
pr Befehlsform :Verb
  rg Wort ol :Verb "!"
ende
```



Sie gibt für ein Verb die Befehlsform des Verbs zusammen mit einem Ausrufezeichen zurück. Z.B. bildet sie aus „denken“ „denke!“ oder „reden“ „rede!“.

Zeichne nun Pfeile von den beiden Trichtern der Maschine zu den entsprechenden Stellen in den Logo-Zeilen. (4 Punkte)

	Test „Syntaxdiagramm“, „Maschinen“	Name:	Datum:	Seite 3
---	---	-------	--------	----------------

Aufgabe 4 (20,5 Punkte)

1. Zeichne ein Syntaxdiagramm, das die 3 Sätze

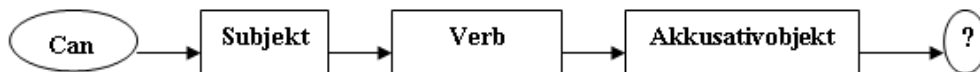
Ben can play football.

Sanjay can sing a song.

Jenny can draw a picture.

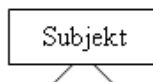
beschreibt. (2,5 Punkte)

2. Bilde aus den 3 Sätzen oben die entsprechenden 3 Fragen, die nach dem Syntaxdiagramm



aufgebaut sind und schreibe die 3 englischen Fragen dann in die freien Zeilen. (3 Punkte)

3. Die Maschine für die Funktion *Subjekt* sieht folgendermaßen aus:

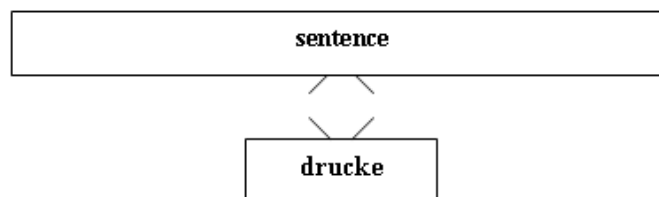


Zeichne die Maschinen für *Verb* und *Akkusativobjekt* auf. (1 Punkt)

	Test „Funktionen“, „Maschinen“, „Logo-Programm“	Name:	Seite 4
---	--	-------	----------------

4. Schreibe *Subjekt*, *Verb* und *Akkusativobjekt* in Logo auf. Mit ihnen sollen später die 3 Beispielsätze und die 3 Fragen aus Aufgabe 4.1. und Aufgabe 4.2 gebildet werden können. (6 Punkte)


5. Zeichne die gesamte Maschine, mit der die Fragen aus Aufgabe 4.2 erzeugt werden können. Ein Teil ist dir schon vorgegeben. (5 Punkte)



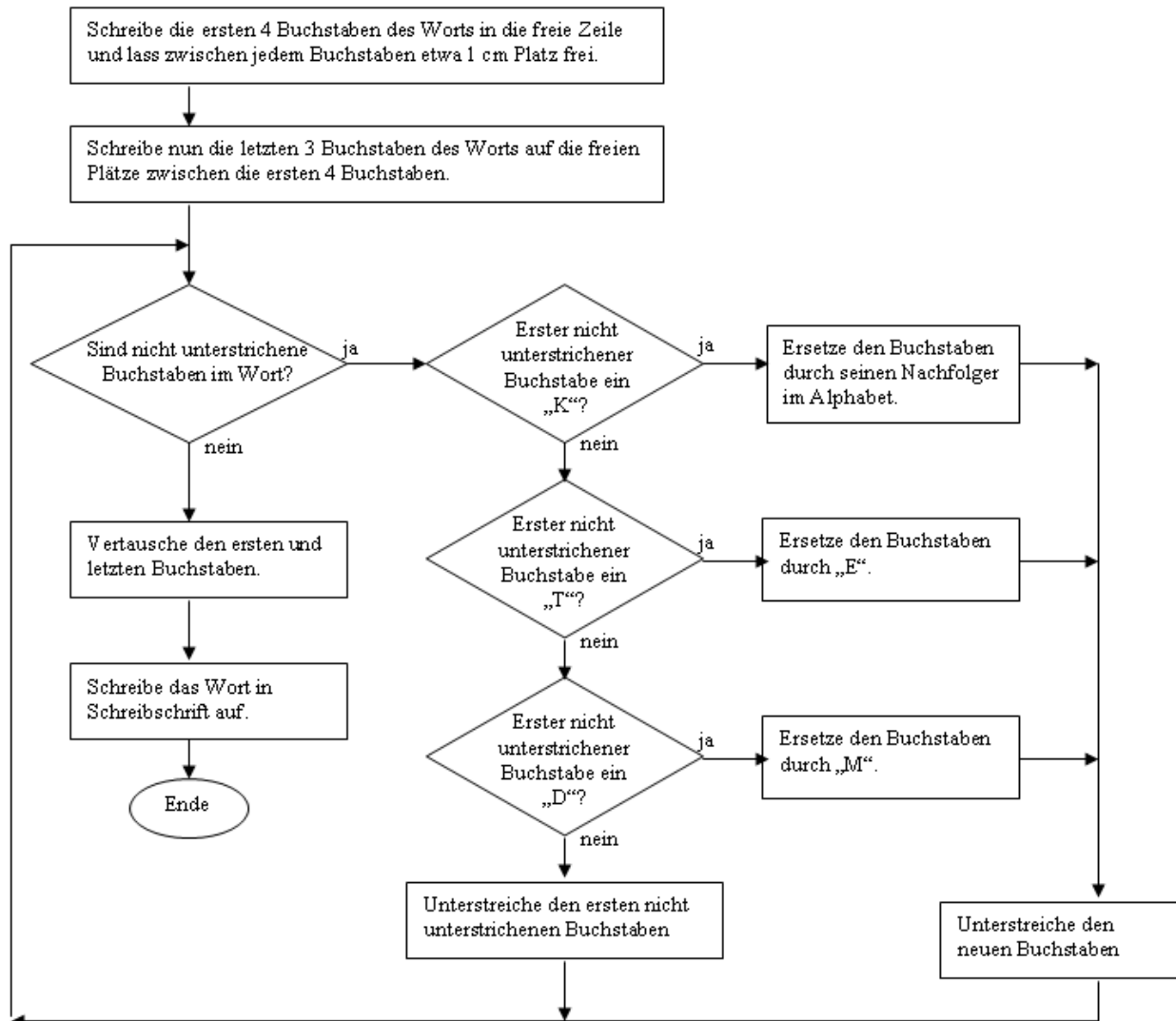
6. Schreibe die komplette Maschine in Logo in die folgende Zeile.

7. Welche Maschinen in Aufgabe 4.5 sind Funktionen? (2 Punkte)

8. Woran erkennst du, dass Maschinen Funktionen sind? (1 Punkt)


	Test „Flussdiagramm“	Name:	Datum:	Seite 5
---	-----------------------------	-------	--------	---------

Aufgabe 5 (8 Punkte)



Knacke den Code für folgende Wörter:

1. GKUFÜKN _____
2. KNSRINA _____
3. NABGKUT _____
4. ZHTSCDR _____

	Test „Flussdiagramm“	Name: _____	Seite 6
---	-----------------------------	-------------	----------------

Zusatzaufgabe (10 Punkte)

1. Verschlüssele du nun das Wort „Schwein“, so dass es mit Hilfe des Flussdiagramms entschlüsselt werden kann. (6 Punkte)

2. Schreibe nun 2 Wörter auf, die man **nicht** mit dem Verfahren aus Aufgabe 5 verschlüsseln und dann entschlüsseln kann. Schreibe den Grund dafür auf. (4 Punkte)

Nutze die freien Zeilen unten, um Zwischenergebnisse für deine verschlüsselten Wörter aufzuschreiben.

1. Wort: _____

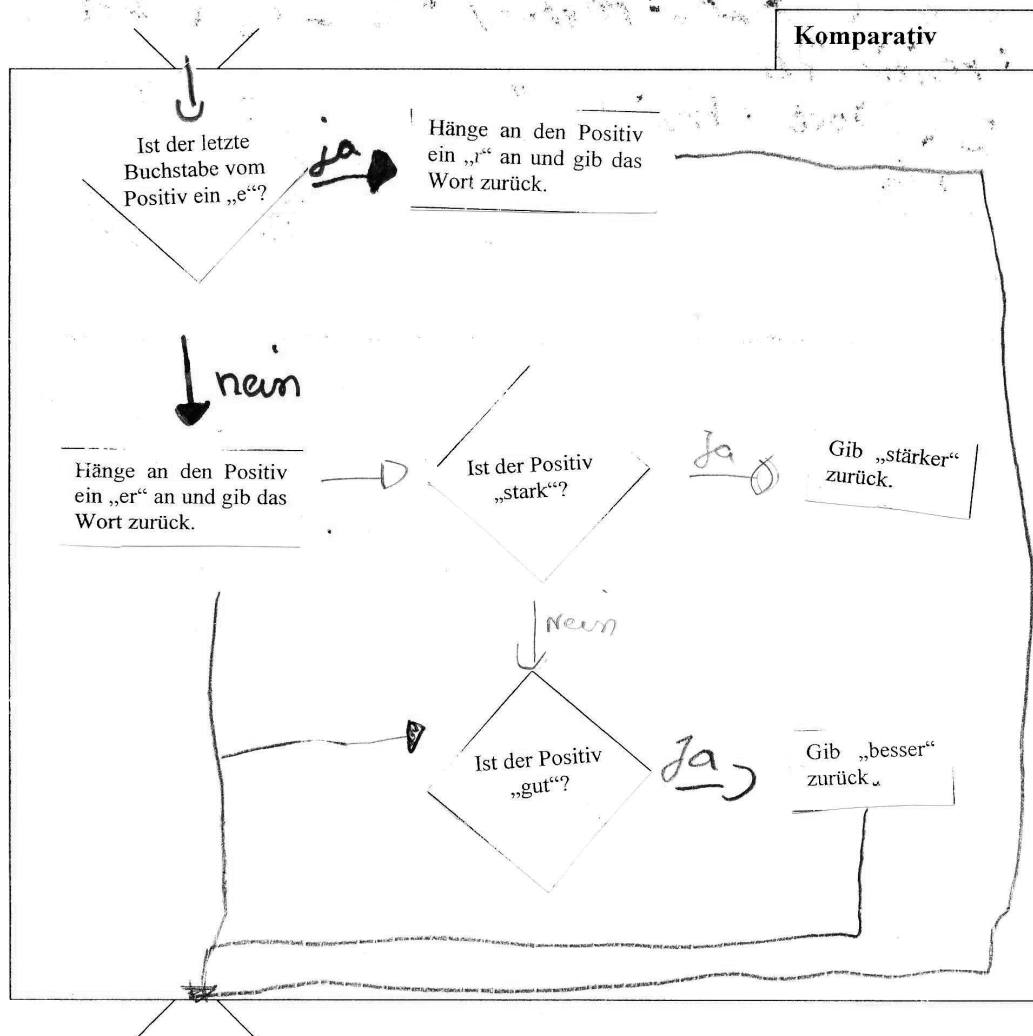
Grund: _____

2. Wort: _____

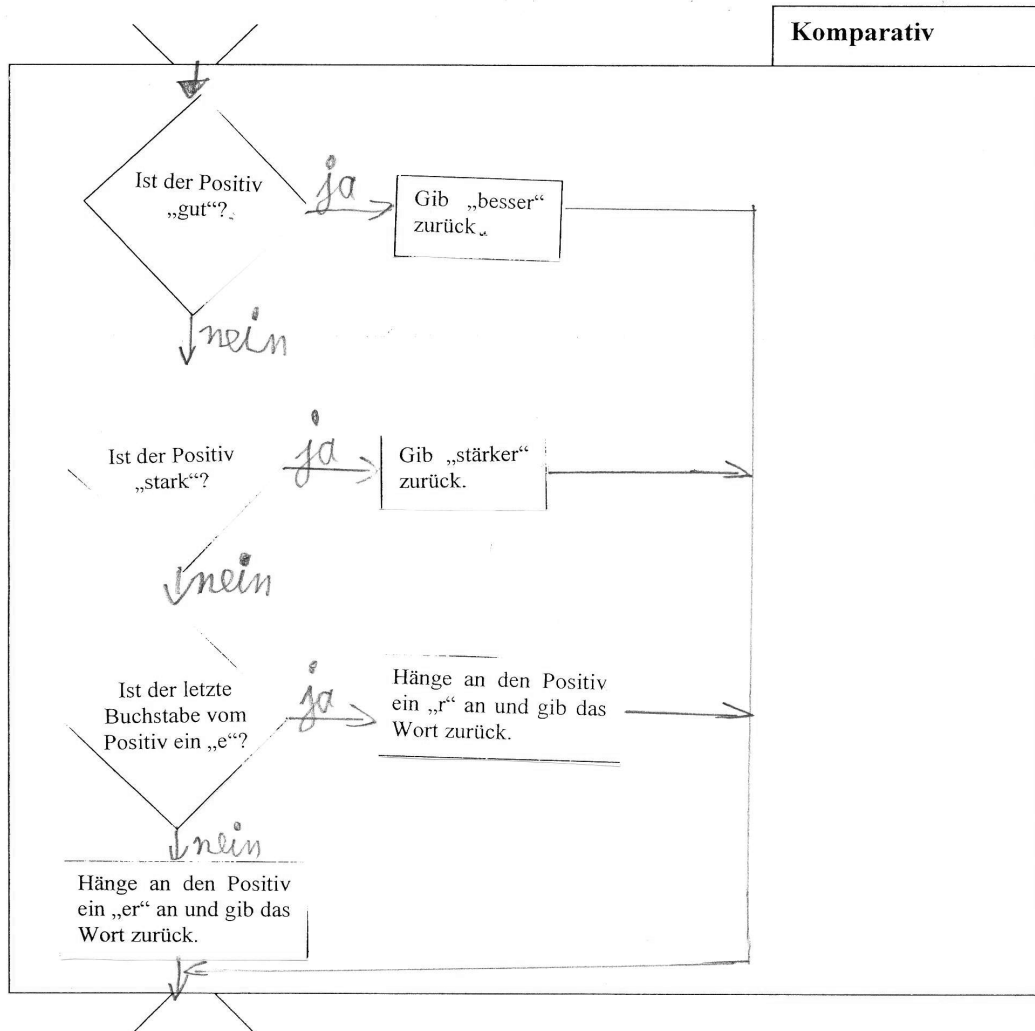
Grund: _____

Anhang H Schülerbeispiele

Programmablaufplan zur Steigerung von Adjektiven – Chaotische Anordnung von Entscheidungen und Handlungen



Programmablaufplan zur Steigerung von Adjektiven – Strukturierte Anordnung von Entscheidungen und Handlungen



Abbildungsverzeichnis

2.1	Entwicklung der Empfehlungen zur informatischen Bildung	6
2.2	<i>Bildungsstandards für Informationstechnische Grundbildung</i> an Realschulen in Baden-Württemberg: Kompetenzen und Inhalte	13
2.3	Fundamentale Ideen der Informatik	15
2.4	Informatik-Turm	16
2.5	Informatische Grundkonzepte als begriffliche und kognitive Grundlage informationstechnischer Kompetenzen und Inhalte	27
3.1	Entwicklung des Unterrichtskonzepts	30
3.2	Zeichnung des Logo-Igels	39
3.3	Quadrat des Logo-Igels	39
3.4	Maschinendarstellung der Befehle Quadrat und Rechteck	40
3.5	Maschinendarstellung der Funktion SIN	40
3.6	Funktionsgeflecht zur Ausgabe eines zufälligen Satzes am Bildschirm	41
3.7	Syntaxdiagramm eines Satzes bestehend aus Subjekt und Prädikat	46
3.8	Klassenzimmer	48
3.9	Computerraum	49
3.10	Benutzeroberfläche von MSW Logo	50
3.11	Benutzeroberfläche des Editors von MSW Logo	50
3.12	Objekthierarchie und Kontextmenü in Imagine Logo	51
3.13	Registerkarte der Attribute eines Igel-Objekts in Imagine Logo	52
3.14	Prozedurhierarchie in Imagine Logo	52
3.15	Oberflächenobjekte in Imagine Logo	53
3.16	Multimediaobjekte in Imagine Logo	53
3.17	Gesamtkonzept	56
4.1	Stadt-Mikrowelt	65
4.2	Abstraktion des Befehls Rechteck	66
4.3	Repräsentation von Tönen als Zahlen	67
4.4	Benutzeroberfläche des englischen Vokabeltrainers	68
4.5	Befehl mit Parameter	69
4.6	Quadratfiguren	70
4.7	Metapher <i>Glücksrad</i>	70
4.8	Benutzeroberfläche für englische Kurzdialoge	71
4.9	UML-Klassendiagramm – Ebene Figuren	72

4.10	Zusammengesetzte Maschinen – Aussagesätze	73
4.11	Zusammengesetzte Maschinen – Abstraktion der Namen in Fragen	74
4.12	Zusammengesetzte Maschinen – Abstraktion der Namen und Eigenschaften in Fragen	74
4.13	Programmablaufplan – Umwandlung römischer Zahlen	76
4.14	Melodieverlauf im Fünfliniensystem	77
4.15	Melodieverlauf als Balkendiagramm	77
4.16	Zusammengesetzte Maschinen – Englische „Is“-Fragen	78
4.17	Programmablaufplan – Bildung des Komparativs	79
4.18	Fadenbilder	80
4.19	Weihnachtskarte	81
4.20	Stellung der Satzglieder in deutschen Sätzen	82
4.21	Syntaxdiagramm – Aussagesätze mit Dativ- und Akkusativobjekten . . .	82
4.22	Melodie im Fünfliniensystem	84
4.23	Maschinen zur Umwandlung von Tonrepräsentationen	84
4.24	Syntaxdiagramm – Has-Sätze	85
4.25	Funktion Potenz als Maschine	85
4.26	Funktion Potenz als Syntaxdiagramm	86
4.27	Syntaxdiagramm – Sätze im <i>present progressive</i>	86
4.28	Zusammengesetzte Maschinen – Sätze im <i>present progressive</i>	87
4.29	Mindmap zu Potenzen	88
4.30	Lernspiel Zehnerpotenzen	89
4.31	Begriffshierarchie	90
4.32	Bildergeschichte	90
4.33	Animation einer Figur	91
4.34	Viertel-Burg	94
4.35	Interaktives Rätsel zum Stammbaum der Ritterfamilie	95
4.36	Rhythmus eines Schülerpaares	95
4.37	Benutzeroberfläche des Autospiels	98
4.38	Rekursive Figur – Vorlage und Lösung einer Schülerin	100
4.39	Benutzeroberfläche des Malprogramms eines Schülerinnenpaares	101
5.1	Untersuchungsplan	104
5.2	Aufgabe zur Modularisierung	105
6.1	Überblick Evaluation	115
6.2	Prozessmodell der induktiven Kategorienbildung	117
6.3	Häufigkeitsverteilung der Punkte im Vor- und Nachtest	144
6.4	Häufigkeitsverteilung der Differenzen der Punkte des Nach- und Vortests	144
6.5	Box-Plots der Differenzen der Punkte des Nach- und Vortests – Geschlecht	145
6.6	Häufigkeitsverteilung der Punkte im Vor- und Nachtest getrennt nach dem Merkmal <i>Geschlecht</i>	146

6.7	Box-Plots der Differenzen der Punkte des Nach- und Vortests – Fähigkeiten	148
6.8	Häufigkeitsverteilungen der Punkte im Vortest getrennt nach den Merkmalen <i>kognitives Gesamtleistungsniveau</i> und <i>sprachgebundenes Denken</i>	148
6.9	Häufigkeitsverteilungen der Punkte im Vortest getrennt nach den Merkmalen <i>zahlengebundenes</i> und <i>formallogisches Denken</i>	149
6.10	Zahlenreihe eines Schülers	151
6.11	Erfolg bei der algorithmischen Formulierung der Zahlenreihe	151
6.12	Erfolg bei der algorithmischen Formulierung der Buchstabenreihe	152
6.13	Erfolg bei der Umwandlung der römischen Zahl MCDXIV in das Dezimalsystem	153
6.14	Erfolg bei der Erstellung des Befehls Quadratfigur	154
6.15	Automatisierte Fadenbilder	155
6.16	Erfolg bei der Erstellung automatisierter Animationen in der Rittergeschichte	156
6.17	Codes der Schülerinnen und Schüler für Daten	157
6.18	Codetypen für Geburtsdaten	157
6.19	Abbildungsvorschrift eines Codes	158
6.20	Erfolg beim Entwurf des Codes für Daten	158
6.21	Nomenklatur der Befehle im Autospiegel	160
6.22	Weihnachtskarte, zusammengesetzt aus den Modulen <i>Kerze</i> und <i>Dreieck</i>	161
6.23	Zusammengesetzte Maschinen – Englische „Is“-Fragen	162
6.24	Erfolg beim Lesen der zusammengesetzten Maschinen englischer „Is“-Fragen	163
6.25	Erfolg bei der iterativen Beschreibung eines Rechtecks	164
6.26	Fehlerhafte Darstellungen rekursiver Kreisfiguren	165
6.27	Erfolg bei der selbstständigen Entwicklung einer rekursiven Figur	165
6.28	Rekursive Figuren der Schülerinnen und Schüler	166
6.29	Rekursive Kreuz- und Rechtecksfiguren	166
6.30	Erfolg bei der Erstellung des Programmablaufplans zur Steigerung von Adjektiven	168
6.31	Erfolg bei der Erstellung eines Syntaxdiagramms für deutsche Sätze unter Vorgabe von Bausteinen	169
6.32	Syntaxdiagramm deutscher Sätze	169
6.33	Erfolg bei der Erstellung eines Syntaxdiagramms für deutsche Sätze ohne Vorgabe von Bausteinen	170
6.34	Erfolg bei der Erstellung eines Syntaxdiagramms für englische Sätze ohne Vorgabe von Bausteinen	170
6.35	Erfolg bei der Bildung der Begriffshierarchie im Vortest	172
6.36	Erfolg bei der Bildung der Begriffshierarchie im Nachtest	172
6.37	Erfolg bei der Ableitung eines Familienstammbaums	173
6.38	Chaotische Anordnung der Familienmitglieder in einem Stammbaum	173
6.39	Erfolg bei der Darstellung der Grammatik für have-Sätze mit Hilfe von Syntaxdiagrammen	174

6.40	Erfolg bei der Darstellung der Grammatik für Genitiv-Sätze mit Hilfe von Syntaxdiagrammen	175
6.41	Erfolg bei der Erstellung der zusammengesetzten Maschinen zu englischen „Is“-Fragen	176
6.42	Veränderung der Einstellung zur Bedeutung von Wissen über den Computer (1)	180
6.43	Veränderung der Einstellung zur Bedeutung von Wissen über den Computer (2)	180
6.44	Veränderung der Einschätzung der Anwendungsmöglichkeiten des Computers	181
6.45	Veränderung der Einschätzung der Auswirkungen der Arbeit mit dem Computer auf die Denkfähigkeiten	182
6.46	Box-Plots der positiven Gefühle bei der Benutzung des Computers	182
6.47	Veränderung des <i>Interesses</i> , der <i>Neugier</i> und der <i>Entspannung</i> bei der Benutzung des Computers	183
6.48	Box-Plots der negativen Gefühle bei der Benutzung des Computers . . .	183
6.49	Veränderung des Gefühls der <i>Zeitverschwendung</i> und der <i>Langeweile</i> bei der Benutzung des Computers	184
6.50	Veränderung der positiven Selbsteinschätzung im Umgang mit dem Computer	185
6.51	Veränderung der negativen Selbsteinschätzung im Umgang mit dem Computer	185
6.52	Veränderung der Einschätzung der Eigenschaften des Computers (1) . . .	186
6.53	Veränderung der Einschätzung der Eigenschaften des Computers (2) . . .	186
6.54	Veränderung der Fehlvorstellungen über Eigenschaften des Computers (1)	187
6.55	Veränderung der Fehlvorstellungen über Eigenschaften des Computers (2)	187
6.56	Veränderung der Einschätzung der Fähigkeiten des Computers	188

Tabellenverzeichnis

2.1	Grundkenntnisse über Computer und Informatik in den Bildungsplänen von 1984	10
2.2	Lehrpläne <i>Informationstechnische Grundbildung</i> und <i>Informationstechnische Grundkenntnisse</i> von 1994	10
2.3	<i>Bildungsstandards für Informationstechnische Grundbildung</i> von 2004 . .	11
2.4	Zusammenhang der Leitlinien informatischer Bildung und informatischer Kompetenz	17
2.5	Kompetenzklassen nach Puhlmann	18
2.6	Positionen der Fachdidaktik Informatik	20
3.1	Programm zur Logo-Fortbildung der Lehrerinnen	59
3.2	Programm zur Logo-Fortbildung der Eltern	60
4.1	Informatische Grundkonzepte eingebettet in Fachinhalte	64
4.2	Informatische Grundkonzepte in der Projektphase	93
4.3	Informatische Grundkonzepte ohne Fachbezug	97
5.1	Aufgaben zu den informatischen Grundkonzepten (1. Teil)	106
5.2	Aufgaben zu den informatischen Grundkonzepten (2. Teil)	107
5.3	Aufgaben zu den informatischen Grundkonzepten (3. Teil)	107
6.1	Schwierigkeitsgrad des Computer- und Fachunterrichts im Vergleich (Schülerinnen und Schüler)	121
6.2	Computer- und Fachunterricht im Vergleich (Schülerinnen und Schüler) .	121
6.3	Frontalphasen, Partnerarbeit am Computer und Theorieunterricht (Lehrerinnen)	122
6.4	Fächerübergreifender Unterricht (Lehrerinnen)	123
6.5	Verschiedene Bearbeitungsformen der Arbeitsblätter (Lehrerinnen)	124
6.6	Änderungswünsche für den Computerunterricht (Schülerinnen und Schüler)	125
6.7	Lernerfolge bezüglich der Fachinhalte im Computerunterricht (Schülerinnen und Schüler)	127
6.8	Lernerfolge bezüglich des Computers und seiner Bedienung (Schülerinnen und Schüler)	127
6.9	Vorteile durch den Besuch der Computer AG (Lehrerinnen)	129
6.10	Mitarbeit im Unterricht (Schülerinnen und Schüler)	131

6.11 Wichtige Arbeiten im Computerunterricht (Schülerinnen und Schüler) . .	132
6.12 Arbeitsformen am Computer (Schülerinnen und Schüler)	133
6.13 Hilfe bei Schwierigkeiten (Schülerinnen und Schüler)	134
6.14 Vor- und Nachteile des Raumwechsels (Lehrerinnen)	135
6.15 Vergleich von Logo und anderen Programmen (Schülerinnen und Schüler)	137
6.16 Vergleich von MSW Logo und Imagine Logo (Schülerinnen und Schüler) .	137
6.17 Einstellungen und Gefühle während der Projektarbeit (Lehrerinnen) . . .	139
6.18 Ursache für fehlende Experimentierfreudigkeit (Lehrerinnen, Schülerinnen und Schüler)	141
6.19 Rohdaten der Tests zu den informatischen Grundkonzepten und des KFT	143
6.20 Verteilung der Mädchen und Jungen in den untersuchten Gruppen	147
6.21 Fehlerkategorien bei der Formulierung der Zahlenreihe als Algorithmus .	151
6.22 Ergebnis der Umwandlung römischer Zahlen in Dezimalzahlen	152
6.23 Fehlende Zwischenschritte bei der Dekodierung von vier Wörtern mit Hilfe eines Programmablaufplans	153
6.24 Fehlerkategorien bei der iterativen Beschreibung des Programmcodes ei- nes Rechtecks	163
6.25 Fehlerkategorien beim Zeichnen der rekursiven Kreisfigur	165
6.26 Beschreibung der rekursiven Kreuzfigur	167
6.27 Zusammenfassung der Auswertung der Unterrichtsmaterialien	179

Literaturverzeichnis

- [Abelson 1982] ABELSON, Harold: A Beginner's Guide to Logo. In: *BYTE* 7(8) (1982), S. 88–112
- [Alimisis & al. 2005] ALIMISIS, Dimitris; KARATRANTOU, Anthi; TACHOS, Nikolaos: Technical school students design and develop robotic gear-based constructions for the transmission of motion. In: GREGORCZYK, Grazyna (Hrsg.); WALAT, Andrzej (Hrsg.); KRANAS, Witold (Hrsg.); BOROWIECKI, Maciej (Hrsg.): *Digital Tools for Lifelong Learning*. Warschau: Proceedings of the Tenth European Logo Conference, 2005, S. 76–86
- [Altermann-Köster & al. 1990] ALTERMANN-KÖSTER, Marita; HOLTAPPELS, Heinz G.; KANDERS, Michael; PFEIFFER, Hermann; DE WITT, Claudia: *Bildung über Computer?* Weinheim und München: Juventa Verlag, 1990
- [Ambrosini & al. 1989] AMBROSINI, G.; BARALDI, D.; BERTOLOTTI, S.; BENATI, G.; BUSSONI, E.; CALABRESE, E.; CANUTI, D.; CASSI, R.; CHIARI, S.; CHIAVARINI, G.; COMPIANI, A.; CORNETTI, A.; CORRADI, C.; DELSANTE, V.; FRIGERI, V.; REGE, C.; RIZZARDI, A.; ROSA, C.; ROSSI, A.; SUPERCHI, A.: Logo and teacher training in Parma's compulsory school. In: SCHUYTEN, G. (Hrsg.); VALCKE, M. (Hrsg.): *Proceedings of the Second European Logo Conference, Gent*, 1989, S. 2–23
- [Baumann 1996] BAUMANN, Rüdiger: *Didaktik der Informatik*. 2., vollständig neu bearbeitete Auflage. Stuttgart: Klett Verlag, 1996
- [Bescherer 2005] BESCHERER, Christine: LoDiC – Learning on Demand in Computing. In: *Proceedings of the 8th IFIP World Conference on Computers in Education 2005, Cape Town*, 2005
- [BLK 1987] BLK: Gesamtkonzept für die informationstechnische Bildung. In: *Materialien zur Bildungsplanung und Forschungsförderung* Bd. 16. Bonn: BLK-Geschäftsstelle, 1987
- [BLK 1995] BLK: Medienerziehung in der Schule. In: *Materialien zur Bildungsplanung und Forschungsförderung* Bd. 44. Bonn: BLK-Geschäftsstelle, 1995
- [BMBF 2003] BMBF: *IT-Ausstattung der allgemein bildenden und berufsbildenden Schulen in Deutschland*. 2003. – URL www.bmbf.de/pub/it-ausstattung_der_schulen_gesamt_2003.pdf – Zugriffsdatum: 03.08.2005

- [Bortz und Döring 2002] BORTZ, Jürgen; DÖRING, Nicola: *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler*. 3. Auflage. Berlin u.a.: Springer-Verlag, 2002
- [Boychev 2003] BOYTCHEV, Pavel: 2. Turtle Metamorphoses (From „FD1“ To 3D Animated Characters). In: *Proceedings of the 9th European Logo Conference – Be creative ... re-inventing technology on education*. Porto, 2003, S. 50–61
- [Brinda 2004] BRINDA, Torsten: *Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II*, Fachbereich 12 – Elektrotechnik und Informatik der Universität Siegen, Dissertation, 2004. – URL www.ub.uni-siegen.de/pub/diss/fb12/2004/brinda/brinda.pdf – Zugriffsdatum: 03.08.2005
- [Chakraborty & al. 1999] CHAKRABORTY, Anit; GRAEBNER, Randy; STOCKY, Tom: *LOGO – A Project History*. 1999. – URL web.mit.edu/6.933/www/LogoFinalPaper.pdf – Zugriffsdatum: 03.08.2005
- [Clark 1985] CLARK, Richard E.: The Importance of Treatment Explication: A Reply to J. Kulik, C-L. Kulik und R. Bangert-Drowns. In: *Journal of Educational Computing Research* 4(1) (1985), S. 398–393
- [Claus und Schwill 2003] CLAUS, Volker; SCHWILL, Andreas: *Schülerduden Informatik*. 4. aktualisierte Auflage. Mannheim u.a.: Dudenverlag, 2003
- [Colella & al. 2001] COLELLA, Vanessa S.; KLOPFER, Eric; RESNICK, Mitchel: *Adventures in Modeling: Exploring Complex, Dynamic Systems with StarLogo*. New York: Teachers College Press, 2001
- [Deutsches PISA-Konsortium 2001] DEUTSCHES PISA-KONSORTIUM (Hrsg.): *PISA 2000*. Opladen: Leske und Budrich Verlag, 2001
- [Edelmann 2000] EDELMANN, Walter: *Lernpsychologie*. 6. vollständig überarbeitete Auflage. Weinheim: Psychologie Verlags Union, 2000
- [Engbring 2003] ENGBRING, Dieter: *Informatik im Herstellungs- und Nutzungskontext – Ein technikbezogener Zugang zur fachübergreifenden Lehre*, Fakultät für Elektrotechnik, Informatik und Mathematik der Universität Paderborn, Dissertation, 2003. – URL ubdata.uni-paderborn.de/ediss/17/2004/engbring/disserta.pdf – Zugriffsdatum: 03.08.2005
- [Espenschied und Mauch 1984] ESPENSCHIED, Wolfgang; MAUCH, Ulrich: *Beispiele zu numerischen Verfahren in der Sekundarstufe I unter Verwendung von Logo*. Didaktisches Zentrum Pädagogische Hochschule Ludwigsburg, 1984. – Informatik und Datenverarbeitung in der Schule – Materialien und Berichte (2)

- [Eyferth & al. 1974] EYFERTH, Klaus; FISCHER, Klaus; KLING, Ulrich; KORTE, Wolfgang; LAUBSCH, Joachim; LÖTHE, Herbert; SCHMIDT, Rainer; SCHULTE, Herbert; WERKHOFER, Konrad: *Computer im Unterricht*. Stuttgart: Ernst Klett Verlag, 1974
- [Farkas 2003] FARKAS, Károly: Logo and native language. Intrinsic procedures of some curves. In: *Proceedings of the 9th European Logo Conference – Be creative ... re-inventing technology on education*. Porto, 2003, S. 69–79
- [Fatke 2003] FATKE, Reinhard: *Jean Piaget. Meine Theorie der geistigen Entwicklung*. Weinheim u.a.: Beltz Verlag, 2003
- [Frey 2003] FREY, Elke: Informatik in der Jahrgangsstufe 6 – ein Bericht aus der Praxis. In: HUBWIESER, Peter (Hrsg.): *Informatische Fachkonzepte im Unterricht*. Bonn: Köllen Druck + Verlag GmbH, 2003, S. 33. – Gesellschaft für Informatik, INFOS 2003 10. GI-Fachtagung Informatik und Schule vom 17.-19. September 2003 in Garching bei München
- [Friedrich 2003] FRIEDRICH, Steffen: Informatik und PISA – vom Wehe zum Wohl der Schulinformatik. In: HUBWIESER, Peter (Hrsg.): *Informatische Fachkonzepte im Unterricht*. Bonn: Köllen Druck + Verlag GmbH, 2003, S. 133–144. – Gesellschaft für Informatik, INFOS 2003 10. GI-Fachtagung Informatik und Schule vom 17.-19. September 2003 in Garching bei München
- [GDM 1981] GDM: *Empfehlungen zur Einbeziehung informatischer Inhalte in den Mathematikunterricht der Sekundarstufe I*. 1981
- [GI 1976] GI: Zielsetzungen und Inhalte des Informatikunterrichts. In: *Zentralblatt für Didaktik der Mathematik* 8(1) (1976), S. 53–60
- [GI 1986] GI: Rahmenempfehlung für die Informatik im Unterricht der Sekundarstufe I. In: *Informatik Spektrum* 9(2) (1986), S. 141–143
- [GI 1993] GI: Empfehlungen für das Fach Informatik in der Sekundarstufe II allgemeinbildender Schulen. In: *Informatik Spektrum* 16(6) (1993), S. 349–356
- [GI 1999] GI: Informatische Bildung und Medienerziehung. In: *Beilage zu LOG IN* 19-6 (1999). – URL www.learn-line.nrw.de/angebote/medienbildung/Foyer/GI/gi_empfehlung.pdf – Zugriffsdatum: 03.08.2005
- [GI 2000] GI: Empfehlungen der Gesellschaft für Informatik e.V. für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen. In: *Informatik Spektrum* 20 (2000), S. 378–382. – URL www.gi-ev.de/informatik/publikationen/gesamtkonzept_26_9_2000.pdf – Zugriffsdatum: 03.08.2005
- [Goldenberg 1982] GOLDENBERG, E. P.: Logo – A Cultural Glossary. In: *BYTE* 7(8) (1982), S. 210–228

- [Gudjons 1997] GUDJONS, Herbert: *Handlungsorientiert lehren und lernen. Schüleraktivierung – Selbständigkeit – Projektarbeit*. 5. Auflage. Bad Heilbrunn/Obb.: Klinkhardt Verlag, 1997
- [Gudjons 1999] GUDJONS, Herbert: *Pädagogisches Grundwissen*. 6. Auflage. Bad Heilbrunn/Obb.: Klinkhardt Verlag, 1999
- [Harper 1989] HARPER, Dennis O.: *LOGO: Theory and Practice*. Pacific Grove, California: Brooks/Cole Publishing Company, 1989
- [Hartmann und Nievergelt 2002] HARTMANN, Werner; NIEVERGELT, Jürg: Informatik und Bildung zwischen Wandel und Beständigkeit. In: *Informatik Spektrum* 25(6) (2002), S. 465–476
- [Harvey 1982] HARVEY, Brian: Why Logo? In: *BYTE* 7(8) (1982), S. 163–193
- [Harvey 1997] HARVEY, Brian: *Computer Science Logo Style – Symbolic Computing*. 2nd edition. Cambridge, Massachusetts: The MIT Press, 1997
- [Hauß & al. 1997] HAUSS, A.; NEUDECKER, W.; TAYLOR, R. J.: *English G 2000 B1, Vorschläge zur Leistungsmessung*. Berlin: Cornelsen Verlag, 1997
- [Heller & al. 2000] HELLER, Kurt A.; GAEDIKE, A.-K.; WEINLÄDER, H.: *KFT 4-13+, Kognitiver Fähigkeits-Test*. Göttingen: Hogrefe, 2000
- [Hense & al. 2001] HENSE, Jan; MANDL, Heinz; GRÄSEL, Cornelia: Problemorientiertes Lernen – Warum der Unterricht mit neuen Medien mehr sein muss als Unterrichten mit neuen Medien. In: *Computer und Unterricht* 44 (2001), S. 6–11
- [Hoppe und Lötke 1984] HOPPE, Heinz U.; LÖTKE, Herbert: *Problemlösen und Programmieren mit LOGO*. Stuttgart: B. G. Teubner Verlag, 1984
- [Huber 1995] HUBER, Ludwig: Individualität zulassen und Kommunikation stiften. In: *Die Deutsche Schule* H.2 (1995), S. 161–182
- [Huber 1998] HUBER, Ludwig: Fächerübergreifender Unterricht – auch auf der Sekundarstufe II? In: DUNCKER, Ludwig (Hrsg.); POPP, Walter (Hrsg.): *Fächerübergreifender Unterricht in der Sekundarstufe I und II*. Bad Heilbrunn/Obb.: Klinkhardt Verlag, 1998, S. 18–33
- [Hubwieser 2001] HUBWIESER, Peter: *Didaktik der Informatik*. 1. Auflage. Berlin Heidelberg: Springer Verlag, 2001
- [Humbert 2001] HUMBERT, Ludger: Informatiklehre – zeitgemäße Ansätze zur nachhaltigen Qualifikation aller Schülerinnen. In: KEIL-SLAWIK, Reinhard (Hrsg.); MARGENHEIM, Johannes (Hrsg.): *Informatikunterricht und Medienbildung*. Bonn: Köllen

- Druck + Verlag GmbH, 2001, S. 121–132. – Gesellschaft für Informatik, INFOS 2001
9. GI-Fachtagung Informatik und Schule vom 17.-20. September 2001 in Paderborn
- [Humbert 2003] HUMBERT, Ludger: *Zur wissenschaftlichen Fundierung der Schulinformatik*, Fachbereich Elektrotechnik und Informatik der Universität Siegen, Dissertation, 2003. – URL www.ub.uni-siegen.de/pub/diss/fb12/2003/humbert/humbert.pdf – Zugriffsdatum: 03.08.2005
- [IFIP Technical Committee on Computer Education 1970] IFIP TECHNICAL COMMITTEE ON COMPUTER EDUCATION: *Computer education for teachers in secondary schools – an outline guide for teachers*. Montvale NJ: AFIPS press, 1970
- [IFIP Technical Committee on Computer Education 1971] IFIP TECHNICAL COMMITTEE ON COMPUTER EDUCATION: *Computer education for teachers in secondary schools – an outline guide*. Montvale NJ: AFIPS press, 1971
- [IFIP Technical Committee on Computer Education 1972] IFIP TECHNICAL COMMITTEE ON COMPUTER EDUCATION: *Computer education for teachers in secondary schools – aims and objectives in teacher training*. Montvale NJ: AFIPS press, 1972
- [Kerres 2001] KERRES, Michael: *Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung*. 2. Auflage. München Wien: Oldenbourg Verlag, 2001
- [Klaudt und Wursthorn 2005] KLAUDT, Dieter; WURSTHORN, Birgit: *Mathematikdidaktik und neue Medien – neue Möglichkeiten und neue Probleme*. In: BESCHERER, Christine (Hrsg.): *Einfluss der neuen Medien auf die Fachdidaktiken – Erfahrungen aus dem Projekt Virtualisierung im Bildungsbereich*. Baltmannsweiler: Schneider Verlag Hohengehren, 2005, S. 173–191
- [Kultusministerium Baden-Württemberg 1984a] KULTUSMINISTERIUM BADEN-WÜRTTEMBERG: *Bildungsplan für das Gymnasium der Normalform*. Villingen-Schwenningen: Neckar-Verlag, 1984 (Kultus und Unterricht)
- [Kultusministerium Baden-Württemberg 1984b] KULTUSMINISTERIUM BADEN-WÜRTTEMBERG: *Bildungsplan für die Realschule*. Villingen-Schwenningen: Neckar-Verlag, 1984 (Kultus und Unterricht)
- [Kultusministerium Baden-Württemberg 1994a] KULTUSMINISTERIUM BADEN-WÜRTTEMBERG: *Bildungsplan für das Gymnasium*. Villingen-Schwenningen: Neckar-Verlag, 1994 (Kultus und Unterricht)
- [Kultusministerium Baden-Württemberg 1994b] KULTUSMINISTERIUM BADEN-WÜRTTEMBERG: *Bildungsplan für die Realschule*. Villingen-Schwenningen: Neckar-Verlag, 1994 (Kultus und Unterricht)

- [Kultusministerium Baden-Württemberg 2004a] KULTUSMINISTERIUM BADEN-WÜRTTEMBERG: *Bildungsplan für das allgemein bildende Gymnasium*. Villingen-Schwenningen: Neckar-Verlag, 2004 (Kultus und Unterricht)
- [Kultusministerium Baden-Württemberg 2004b] KULTUSMINISTERIUM BADEN-WÜRTTEMBERG: *Bildungsplan für die Realschule*. Villingen-Schwenningen: Neckar-Verlag, 2004 (Kultus und Unterricht)
- [Lawler 1982] LAWLER, Robert W.: Designing Computer-Based Microworlds. In: *BYTE* 7(8) (1982), S. 138–160
- [Löthe 2004] LÖTHE, Herbert: Responsible and well-devised computer use – standards for computer as working and learning tool in elementary education. In: *Learning for 21st century: What really matters?*, 2004, S. 41–42. – International workshop of IFIP WG 3.5, 28. June - 2. July 2004 in Budapest
- [Löthe & al. 1987] LÖTHE, Herbert; WÖLPERT, Heinrich; WOLPERT, Sibylle: *Raumigel – Dokumentation einer Arbeitsgemeinschaft in Klasse 9 Realschule*. Didaktisches Zentrum Pädagogische Hochschule Ludwigsburg, 1987. – Informatik und Datenverarbeitung in der Schule – Materialien und Berichte (14)
- [Masterson 1985] MASTERSON, Fred A.: Evaluating Logo: A Case Study in Requirements for Student Programming Languages. In: *Computers in the Schools* 2(2/3) (1985), S. 179–195
- [Mayer 2004] MAYER, Richard E.: Should There Be a Three-Strikes Rule Against Pure Discovery Learning? The Case for Guided Methods of Instruction. In: *American Psychologist* 59(1) (2004), S. 14–19
- [Mayring 2003] MAYRING, Philipp: *Qualitative Inhaltsanalyse*. 8. Auflage. Weinheim und Basel: Beltz Verlag, 2003
- [Mittermeir 2003] MITTERMEIR, Roland T.: Wofür lehren wir Informatik! In: *LOG IN* 122/123 (2003), S. 43–45
- [MNU 1985a] MNU: Empfehlungen und Überlegungen zur Gestaltung von Lehrplänen für den Computer-Einsatz im Unterricht der allgemeinbildenden Schulen. In: *Der mathematische und naturwissenschaftliche Unterricht* Bd. 38(4). Bonn: Ferd. Dümmlers Verlag, 1985
- [MNU 1985b] MNU: Empfehlungen zur Gestaltung von Lehrplänen für die informationstechnische Bildung in der Sekundarstufe I bzw. II und für den Computer-Einsatz im Mathematikunterricht der Sekundarstufe II. In: *Der mathematische und naturwissenschaftliche Unterricht* Bd. 39(2). Bonn: Ferd. Dümmlers Verlag, 1985

- [Modrow 2003] MODROW, Eckart: *Pragmatischer Konstruktivismus und fundamentale Ideen als Leitlinien der Curriculumentwicklung am Beispiel der theoretischen und technischen Informatik*, Mathematisch-Naturwissenschaftlich-Technische Fakultät der Martin-Luther-Universität Halle-Wittenberg, Dissertation, 2003. – URL sundoc.bibliothek.uni-halle.de/diss-online/03/03H066/t1.pdf – Zugriffsdatum: 03.08.2005
- [Moegling 1998] MOEGLING, Klaus: Fächerübergreifender Unterricht. In: MOEGLING, Klaus (Hrsg.): *Fächerübergreifender Unterricht – Wege ganzheitlichen Lernens in der Schule*. Bad Heilbrunn/Obb.: Klinkhardt Verlag, 1998. – URL www.sowi-online.de/reader/integration/09beitrag.htm – Zugriffsdatum: 03.08.2005
- [Nievergelt 1995] NIEVERGELT, Jürg: Welchen Wert haben theoretische Grundlagen für die Berufspraxis? Gedanken zum Fundament des Informatikturns. In: *Informatik Spektrum* 18(6) (1995), S. 342–344
- [Papert 1982] PAPERT, Seymour: *Mindstorms – Kinder, Computer und Neues Lernen*. Basel: Birkhäuser Verlag, 1982
- [Papert 1998] PAPERT, Seymour: *Die vernetzte Familie*. Stuttgart: Kreuz Verlag, 1998
- [Papert 1999] PAPERT, Seymour: What is Logo? Who Needs It? In: *Logo Philosophy and Implementation*. Logo Computer Systems Inc., 1999, S. V–XVI. – URL www.microworlds.com/company/philosophy.pdf – Zugriffsdatum: 03.08.2005
- [Puhlmann 2003] PUHLMANN, Hermann: Informatische Literalität nach dem PISA-Muster. In: HUBWIESER, Peter (Hrsg.): *Informatische Fachkonzepte im Unterricht*. Bonn: Köllen Druck + Verlag GmbH, 2003, S. 145–154. – Gesellschaft für Informatik, INFOS 2003 10. GI-Fachtagung Informatik und Schule vom 17.-19. September 2003 in Garching bei München
- [Quehl 1985] QUEHL, Werner: *Logo in der Sekundarstufe I – Kursunterlagen*. Didaktisches Zentrum Pädagogische Hochschule Ludwigsburg, 1985. – Informatik und Datenverarbeitung in der Schule – Materialien und Berichte (4)
- [Reichert 2003] REICHERT, Raimond: *Theory of Computation as a Vehicle for Teaching Fundamental Concepts of Computer Science*, ETH Zürich, Dissertation, 2003. – URL e-collection.ethbib.ethz.ch/ecol-pool/diss/abstracts/p15035.pdf – Zugriffsdatum: 03.08.2005
- [Reichert & al. 2004] REICHERT, Raimond; NIEVERGELT, Jürg; HARTMANN, Werner: *Programmieren mit Kara. Ein spielender Zugang zur Informatik*. Berlin Heidelberg: Springer Verlag, 2004

- [Resnick 1996] RESNICK, Mitchel: New Paradigms for Computing, New Paradigms for Thinking. In: KAFAI, Yasmin (Hrsg.); RESNICK, Mitchel (Hrsg.): *Constructionism in Practice. Designing, Thinking, and Learning in a Digital World*. Mahwah, New Jersey: Lawrence Erlbaum Associates, 1996, S. 255–267
- [Resnick 2001] RESNICK, Mitchel: *Turtles, Termites, and Traffic Jams. Explorations in Massively Parallel Microworlds*. MIT Press, 2001
- [Schnurer & al. 2003] SCHNURER, Katharina; STARK, Robin; MANDL, Heinz: Auf dem Weg in eine neue Lehr-Lern-Kultur. Gestaltung problemorientierter Lernumgebungen. In: *Erziehungswissenschaft und Beruf* 2 (2003), S. 147–160
- [Schrage 1994] SCHRAGE, Michael: Beware the Computer Technocrats. Hardware Won't Educate Our Kids. In: *Educational Media and Technology Yearbook*. Englewood, 1994 (20), S. 64–65
- [Schubert und Schwill 2004] SCHUBERT, Sigrid; SCHWILL, Andreas: *Didaktik der Informatik*. Heidelberg u.a.: Spektrum Akademischer Verlag, 2004
- [Schulmeister 1996] SCHULMEISTER, Rolf: *Grundlagen hypermedialer Lernsysteme. Theorie – Didaktik – Design*. Bonn u.a.: Addison-Wesley, 1996
- [Schulte 2003] SCHULTE, Carsten: *Lehr-Lernprozesse im Informatik-Anfangsunterricht – Theoriegeleitete Entwicklung und Evaluation eines Unterrichtskonzepts zur Objektorientierung in der Sekundarstufe II*, Fakultät für Elektrotechnik, Informatik und Mathematik der Universität Paderborn, Dissertation, 2003. – URL www.informatica-didactica.de/Examensarbeiten/Schulte2003.pdf – Zugriffsdatum: 03.08.2005
- [Schwill 2001] SCHWILL, Andreas: Ab wann kann man mit Kindern Informatik machen? – Eine Studie über informatische Fähigkeiten von Kindern. In: KEIL-SLAWIK, Reinhard (Hrsg.); MAGENHEIM, Johannes (Hrsg.): *Informatikunterricht und Medienbildung*. Bonn: Köllen Druck + Verlag GmbH, 2001, S. 13–30. – Gesellschaft für Informatik, INFOS 2001 9. GI-Fachtagung Informatik und Schule vom 17.-20. September 2001 in Paderborn
- [Shimabukuro 1988] SHIMABUKURO, Gini: *Thinking in Logo – A Sourcebook for Teachers of Primary Students*. Addison-Wesley, 1988
- [Sipitakiat und Cavallo 2003] SIPITAKIAT, Arnan; CAVALLO, David P.: Digital Technology for Conviviality: Making the Most of Learners' Energy and Imagination. In: *Proceedings of the 9th European Logo Conference – Be creative ... re-inventing technology on education*. Porto, 2003, S. 264–273

- [Sperber 2002] SPERBER, Michael: *Programmieren für alle, Projekt „DeinPProgram“*. 2002. – URL www.deinprogramm.de/overview/overview.html – Zugriffsdatum: 03.08.2005
- [Stritzky 1995] STRITZKY, Regine V.: *Informationstechnische Grundbildung in der Schule: eine empirische Untersuchung zu Voraussetzungen und Wirkungen eines neuen Lernangebots für die Sekundarstufe I*. Münster u.a.: Waxmann Verlag, 1995
- [Valente 1997] VALENTE, José A.: Contextualizing continuous education in logo via internet. In: TURCSANYI-SZABO, M. (Hrsg.): *Learning and Exploring with Logo*. Budapest: Proceedings of the Sixth European Logo Conference, 1997, S. 70–79. – URL eurologo.web.elte.hu/lectures/valente.htm – Zugriffsdatum: 03.08.2005
- [Watt und Watt 1986] WATT, Molly; WATT, Daniel: *Teaching With Logo – Building Blocks for Learning*. Addison-Wesley Verlag, 1986
- [Weber 1988] WEBER, Joachim: *Grafikwerkzeuge – Implementation und Einsatz in Klasse 8 Realschule*. Didaktisches Zentrum Pädagogische Hochschule Ludwigsburg, 1988. – Informatik und Datenverarbeitung in der Schule – Materialien und Berichte (16)
- [Wedekind & al. 2004a] WEDEKIND, Hartmut; ORTNER, Erich; INHETVEEN, Rüdiger: Informatik als Grundbildung. In: *Informatik Spektrum* 27(2) (2004), S. 172–180
- [Wedekind & al. 2004b] WEDEKIND, Hartmut; ORTNER, Erich; INHETVEEN, Rüdiger: Informatik als Grundbildung. Teil II: Bildung von Elementarsätzen. In: *Informatik Spektrum* 27(3) (2004), S. 265–272
- [Wedekind & al. 2004c] WEDEKIND, Hartmut; ORTNER, Erich; INHETVEEN, Rüdiger: Informatik als Grundbildung. Teil III: Gleichheit und Abstraktion. In: *Informatik Spektrum* 27(4) (2004), S. 337–341
- [Wedekind & al. 2004d] WEDEKIND, Hartmut; ORTNER, Erich; INHETVEEN, Rüdiger: Informatik als Grundbildung. Teil IV: Objektsprache/Metasprache. In: *Informatik Spektrum* 27(5) (2004), S. 459–466
- [Wedekind & al. 2005] WEDEKIND, Hartmut; ORTNER, Erich; INHETVEEN, Rüdiger: Informatik als Grundbildung. Teil VI: Logik und Geltungssicherung. In: *Informatik Spektrum* 28(1) (2005), S. 48–52
- [Wessels 1994] WESSELS, Michael G.: *Kognitive Psychologie*. 3. Auflage. New York: Harper & Row (UTB für Wissenschaft), 1994

- [Wursthorn 2005a] WURSTHORN, Birgit: Erwerb informatischer Grundkonzepte durch Spielen. In: ENGEL, J. (Hrsg.); VOGEL, R. (Hrsg.); WESSOLOWSKI, S. (Hrsg.): *Strukturieren – Modellieren – Kommunizieren; Leitbilder mathematischer und informatischer Aktivitäten*. Hildesheim, Berlin: Franzbecker Verlag, 2005, S. 313–326
- [Wursthorn 2005b] WURSTHORN, Birgit: Fundamental concepts of computer science in a Logo-environment. In: GREGORCZYK, Grazyna (Hrsg.); WALAT, Andrzej (Hrsg.); KRANAS, Witold (Hrsg.); BOROWIECKI, Maciej (Hrsg.): *Digital Tools for Lifelong Learning*. Warschau: Proceedings of the Tenth European Logo Conference, 2005, S. 219–227
- [Wursthorn 2005c] WURSTHORN, Birgit: *Informatische Grundkonzepte zu Beginn der Sekundarstufe I*. 2005. – wird im Tagungsband der Konferenz INFOS 2005 (Dresden) veröffentlicht
- [Zöfel 2002] ZÖFEL, Peter: *Statistik verstehen*. München: Addison-Wesley Verlag, 2002
- [Ziegenbalg 1985] ZIEGENBALG, Jochen: *Programmieren lernen mit Logo*. München u.a.: Carl Hanser Verlag, 1985